LOBOT: Low-Cost, Self-Contained Localization of Small-Sized Ground Robotic Vehicles

Guoxing Zhan and Weisong Shi, *Senior Member, IEEE*

Abstract—It is often important to obtain the real-time location of a small-sized ground robotic vehicle when it performs autonomous tasks either indoors or outdoors. We propose and implement LOBOT, a low-cost, self-contained localization system for small-sized ground robotic vehicles. LOBOT provides accurate real-time, three-dimensional positions in both indoor and outdoor environments. Unlike other localization schemes, LOBOT does not require external reference facilities, expensive hardware, careful tuning or strict calibration, and is capable of operating under various indoor and outdoor environments. LOBOT identifies the local relative movement through a set of integrated inexpensive sensors and well corrects the localization drift by infrequent GPS-augmentation. Our empirical experiments in various temporal and spatial scales show that LOBOT keeps the positioning error well under an accepted threshold.

Index Terms—Localization, robot, sensor, GPS.

1 Introduction

Small-sized ground robotic vehicles have great potential to be deployed in situations that are either uncomfortable for humans or simply too tedious. For example, a robot may become part of industrial operations, or become part of a senior citizen's life, or become a tour guide for an exhibition center. The robot is kept as small as possible to allow access through narrow passageways such as a tunnel. To fulfill these missions, the robotic vehicle often has to obtain its accurate localization in real time. Considering the difficulty or impossibility in frequent calibration or the management of external facilities, it is desirable to have a self-contained positioning system for the robot: ideally, the localization system should be completely integrated onto the robot instead of requiring external facilities to obtain the position; the system should work indoors and outdoors without any human involvement such as manual calibration or management. Meanwhile, the cost is expected to be as low as possible.

There exist various localization schemes for ground robotic vehicles. These techniques normally utilize GPS, inertial sensors, radio signals, or visual processing. GPS often becomes inoperable in certain environments such as indoors or in wild forests. Additionally, the GPS operations consume power quickly. As an alternative, a localization system may use various waves including electromagnetic waves of various frequency (e.g., common WiFi radio, Ultra-wideband [1], RFID radio [2], Infrared [3]), laser beam [4], and ultrasound [5]. The radio-based positioning is among the most popular techniques. This technology requires a set of external devices to generate or receive radio signal; as the reference nodes, these external devices should have known positions. The accuracy of the radio-based positioning strongly depends on the proper calibration of the reference devices and the target node [6], [7] as well as a friendly radio environment. Maintaining such a positioning system can be costly and difficult in terms of additional hardware [8], [9], [10], intensive tuning [11], and environmental management. It is also vulnerable to interference from other signals, thus affecting the accuracy of positioning.

Another category of solutions is vision techniques for mobile robot navigation [12]. Generally, these techniques heavily rely on sophisticated techniques on the recognition of an object or shape from images and often have restricted spatial and visional requirements. The performance usually strongly depends on the environment in which the robot operates and the localization suffers frequent failure. Additionally, they may require a known map of the environment. Overall, the vision-based positioning is relatively costly and difficult to implement or maintain.

Additionally, inertial sensors are often used in positioning or navigation systems to detect movement [13], [14], [15], [16], [17]. Different than the radio-based and the vision-based techniques, the operation of inertial sensors is independent of external features in the environment and they do not need an external reference. The inertial sensors mainly comprise accelerometers and gyroscopes (gyros). An accelerometer measures specific force and a gyroscope measures angular rate. Many inertial systems often require extremely accurate inertial sensors to maintain accuracy, which often causes high cost and calibration difficulty. Being widely-available and inexpensive, the accelerometer is often perceived as a solution for localization. The accelerometer-based positioning schemes generally use the following formula to derive distance from a given acceleration a:

- *G. Zhan is a Ph.D. candidate in the Department of Computer Science, Wayne State University, Detroit, MI, 48202.*
  E-mail: gxzhan@wayne.edu
- *W. Shi is an Associate Professor of Computer Science at Wayne State University.*
  E-mail: weisong@wayne.edu

$s(t) = \int\int a(t)dtdt$. In spite of being theoretically well founded, empirically, the double integral is likely to cause cumulative error. The methods proposed to correct this error often have not been thoroughly evaluated yet.

To resolve the aforementioned issues, we propose LOBOT, a low-cost, self-contained localization system for the small-sized ground robotic vehicle. LOBOT identifies the real-time localization through a set of self-integrated inexpensive sensors including an accelerometer, a magnetic field sensor, several motor rotation sensors, and infrequent GPS-augmentation. It detects local relative position with a combination of the accelerometer, the magnetic field sensor and the motor rotation sensors. LOBOT infrequently invokes the GPS-augmentation to assist in identifying global location and correcting drifting errors. LOBOT can be applied to both indoor and outdoor environments. These extra sensing devices including the GPS receiver are integrated onto the ground robotic vehicle and only induce a limited cost to the vehicle. LOBOT does not require any external facilities or prior information and it virtually needs no effort of external maintenance. LOBOT is free of many common requirements or issues raised in other localization schemes such as radio-based schemes and vision-technique-based schemes, such as expensive hardware, external reference facilities, careful tuning or strict calibration, and prior map information. It also has significant improvement in location precision over the purely-accelerometer-based approach. We developed a prototype of the LOBOT system and conducted various field evaluation. The empirical results indicate the satisfactory performance of LOBOT.

The rest of the paper is organized as follows: the detailed mechanism of LOBOT is described in Section 2; the implementation and empirical evaluation of LOBOT are given in Section 3; the conclusions are presented in Section 4. An appendix is also presented as supplemental material to help illustrate the different reference frames.

2 The Design of LOBOT

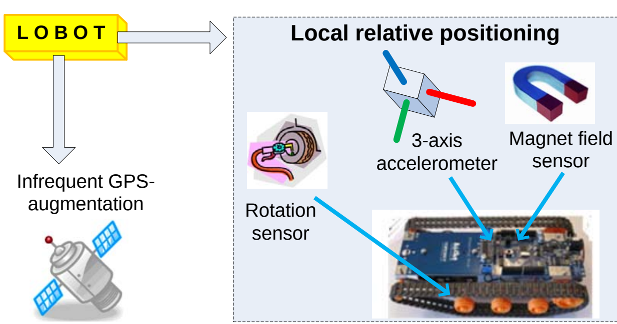


Fig. 1. The design of LOBOT.

LOBOT localizes a robotic vehicle with a hybrid approach consisting of infrequent absolute positioning through a GPS receiver and local relative positioning based on a 3D accelerometer, a magnetic field sensor and several motor rotation sensors (Fig. 1). All these sensors are installed on the robotic vehicle. The motor rotation sensors are to detect the rotational movement of the motors and thus infer the travel distance of the robot. An embedded microcontroller inside the robot vehicle takes central control of these sensors and is also responsible for computing the current absolute position. LOBOT infrequently uses GPS to obtain an absolute position and utilizes the accelerometer, the magnetic field sensor and the motor rotation sensors to measure local relative movement since the last known absolute position through GPS. With the GPS data, correction is performed to reduce the cumulative error from the local relative positioning component. The infrequent use of GPS reduces the dependence on the environmental impact, e.g., a small area without GPS signal. As a matter of fact, even if GPS is available, LOBOT may still only uses the local relative component over a short time period instead of GPS because GPS is known to have error of up to 20m while the local relative component has much lower error over a short time elapse. Additionally, the infrequent use of GPS saves electric power.

The local relative positioning component measures the instantaneous three-dimensional moving direction through both the accelerometer and the magnetic field sensor. It also measures the momentary travel distance for every small amount of time elapse through the rotation sensors attached to the vehicle motors. With the moving direction data together with the momentary travel distance, we can obtain an estimate of the movement vector. This seemingly straightforward strategy, however, has encountered a few major technical issues that arise in practical applications. One lies in the distinction between the world reference system and the on-board relative reference system. Another factor that impacts the localization practice is the way the robotic vehicle operates the motors to move. A further complication comes from the cumulative error.

The overall procedure for LOBOT to decide the position is illustrated by Fig. 2. Roughly, the local relative positioning infers the momentary moving orientation (Subsection 2.2) and estimates the momentary travel distance (Subsection 2.3), with the aid of the accelerometer, the magnetic sensor, and the rotation sensors. The local relative positioning accumulates these momentary estimates to compute the position of the vehicle at any time. Over certain time elapse, the infrequent GPS-augmentation is conducted and is used to perform drift correction (Subsection 2.4) so as to obtain better position estimate.

LOBOT is a low-cost, self-contained system. All the necessary hardware devices needed to perform the positioning are a GPS receiver, a 3D accelerometer, a magnetic field sensor, and several motor rotation sensors. LOBOT only needs the commodity versions of these devices that come with moderate precision and low prices. For ease of development, our prototype uses a GPS

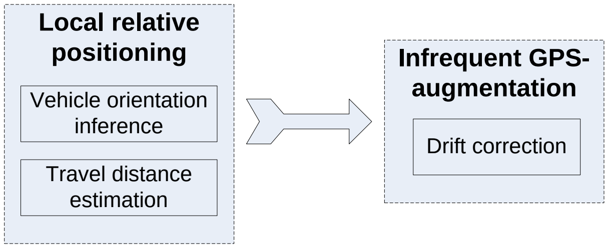


Fig. 2. The overall procedure of LOBOT.

receiver, a 3D accelerometer, a magnetic field sensor from an unlocked HTC Legend smartphone that is sold at no more than $300 at the time of this writing. The motor rotation sensors used in this prototype is obtained from a brand of hobby servo motor that sells at $20. Given a complete circuit design, the actual cost of manufacturing a microcontroller chip integrating all these raw sensors (including the GPS receiver) can very likely be brought down to well under $100 per set. Additionally, all these sensing devices including the GPS receiver can be well powered by the battery of the HTC legend smartphone. Compared with the intense power needed to drive a robotic vehicle, these sensing devices induce only limited overhead in the power consumption. Thus, LOBOT is a low-cost system. The self-containedness of LOBOT is reflected in two aspects: virtually no requirement of external devices or external facility management; no prior information needed. All the necessary devices are attached to the body of the robotic vehicle that we need to localize. Except for GPS, LOBOT does not require any external devices (e.g., a reference anchor point). The GPS satellite network is maintained by official organizations and thus the use of a GPS receiver virtually needs no effort to maintain external facilities. Unlike many positioning schemes based on vision recognition techniques, LOBOT does not require prior information of the environment either.

2.1 Reference Frames

To determine the current moving orientation, we will first need to make a choice on the reference frame. The direction is expressed in a coordinate system relative to the reference frame chosen. In the appendix presented as supplemental material, we present more intuitive illustration of the reference frames used. Here we briefly cover the definition of the reference frames and their meanings. We adopt a right-handed orthogonal reference frame, *LOBOTFrame*$\{X_L, Y_L, Z_L\}$ as follows: the Y axis is parallel to the magnetic field of the earth and points towards the magnetic north pole; the Z axis points towards the sky and is parallel to the gravitational force; the X axis is defined as the outer vector product of a unit vector of Y and that of Z so that $\{X_L, Y_L, Z_L\}$ defines a right-handed orthogonal reference frame. For the purpose of measuring relative movement, the choice of the origin does not affect our result and thus we omit the origin when describing the reference frames. Additionally, we assume that in an area being explored by the robot the directions of both the gravitational force and the earth's magnetic field are constant. As a matter of fact, the gravitational direction rarely changes in a city-magnitude area. The change of the earth's magnetic field direction in such an area is usually also negligible without the existence of another strong magnetic field. If the strength of another magnetic field is so strong that it causes a noticeable difference on the readings of the magnetic sensor, LOBOT will switch to the pure GPS-based mode if the GPS service is available. Thus, we have a well-defined reference frame *LOBOTFrame* for measuring the relative movement of the vehicle. Roughly, the X axis is tangential to the ground at the robot's current location and points east; the Y axis is tangential to the ground and points north (it is slightly different than the magnetic north); the Z axis roughly points towards the sky and is perpendicular to the ground.

Before introducing how to determine the robot's moving orientation, we first show three other closely related right-handed orthogonal reference frames. Unlike *LOBOTFrame*, these frames change as the robot moves. The first one is the reference frame relative to the rigid body of the robot, which we name *VehicleBodyFrame*. *VehicleBodyFrame* is not a static frame when the vehicle moves. Specifically, *VehicleBodyFrame* is a right-handed orthogonal reference frame $\{X_V, Y_V, Z_V\}$, described as follows: the Y axis is parallel to the lines connecting the centers of a motor and another motor right behind it, and points to the front; the Z axis points towards the sky and is perpendicular to the surface containing all the centers of the motors; the X axis is defined as the outer vector product of a unit vector of the Y axis and that of the Z axis so that $\{X_V, Y_V, Z_V\}$ defines a right-handed orthogonal reference frame (the X axis points to the right side of the vehicle).

Another relative reference frame, denoted as *AccelerometerBodyFrame*, is also a right-handed orthogonal reference frame $\{X_A, Y_A, Z_A\}$ on which the accelerometer reading is based. Usually the 3D reading from an accelerometer indicates how the measured acceleration is decomposed into these three axis directions. This reference frame is relative to the circuit board of the accelerometer and is defined by the manufacturer. Two of the axes are often parallel to the circuit board. Similarly, the last reference frame which we name as *MagneticSensorBodyFrame*, is another right-handed orthogonal relative reference frame $\{X_M, Y_M, Z_M\}$ on which the magnetic sensor reading is based. Note that *VehicleBodyFrame*, *AccelerometerBodyFrame* and *MagneticSensorBodyFrame* may all change when the vehicle moves; however, a fixed installation ensures inherent unchanged relations between *VehicleBodyFrame* and the two latter frames and such relations can be decided during installation.

2.2 Inferring Orientation of Robotic Vehicle

Now we describe how LOBOT infers the current instantaneous moving direction of the robotic vehicle relative to *LOBOTFrame*, which is a static frame (relative to the earth). Denote the unit vectors along the axes of each reference frame (normalized basis vector) as in Table 1. To infer the orientation of the vehicle, it is enough to express $\{\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V\}$ in terms of $\{\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L\}$. Given the gravitational acceleration vector g, then

TABLE 1
Reference frames and their normalized basis vectors

Frame	Normalized basis vectors
LOBOTFrame	$\{\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L\}$
VehicleBodyFrame	$\{\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V\}$
AccelerometerBodyFrame	$\{\widehat{X}_A, \widehat{Y}_A, \widehat{Z}_A\}$
MagneticSensorBodyFrame	$\{\widehat{X}_M, \widehat{Y}_M, \widehat{Z}_M\}$

$$\widehat{Z}_L = -\frac{g}{\|g\|} \tag{1}$$

Let the normalized accelerometer reading be (a_1, a_2, a_3) relative to *AccelerometerBodyFrame*. Then

$$\widehat{Z}_L = -\frac{g}{\|g\|} = a_1 \cdot \widehat{X}_A + a_2 \cdot \widehat{Y}_A + a_3 \cdot \widehat{Z}_A \tag{2}$$

Similarly, given the normalized reading (m_1, m_2, m_3) from the magnetic sensor, we have

$$\widehat{Y}_L = m_1 \cdot \widehat{X}_M + m_2 \cdot \widehat{Y}_M + m_3 \cdot \widehat{Z}_M \tag{3}$$

Let T_{AV} be the transformation matrix between *AccelerometerBodyFrame* and *VehicleBodyFrame*, T_{MV} be the transformation matrix between *MagneticSensorBodyFrame* and *VehicleBodyFrame*, so that

$$\begin{aligned}(\widehat{X}_A, \widehat{Y}_A, \widehat{Z}_A) &= (\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V) \cdot T_{AV} &(4)\\ (\widehat{X}_M, \widehat{Y}_M, \widehat{Z}_M) &= (\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V) \cdot T_{MV} &(5)\end{aligned}$$

Thus, we have the following equations:

$$\begin{aligned}\widehat{Z}_L &= (a_1, a_2, a_3) \cdot (\widehat{X}_A, \widehat{Y}_A, \widehat{Z}_A)' &(6)\\ &= (a_1, a_2, a_3) \cdot T_{AV}' \cdot (\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V)' &(7)\\ \widehat{Y}_L &= (m_1, m_2, m_3) \cdot (\widehat{X}_M, \widehat{Y}_M, \widehat{Z}_M)' &(8)\\ &= (m_1, m_2, m_3) \cdot T_{MV}' \cdot (\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V)' &(9)\end{aligned}$$

Now, we are able to construct a special orthogonal matrix as the transformation matrix T_{LV} between *LOBOTFrame* and *VehicleBodyFrame* as follows: the second column vector of T_{LV} is:

$$((m_1, m_2, m_3) \cdot T_{MV}')' = T_{MV} \cdot (m_1, m_2, m_3)' \tag{10}$$

The third column vector is:

$$((a_1, a_2, a_3) \cdot T_{AV}')' = T_{AV} \cdot (a_1, a_2, a_3)' \tag{11}$$

The first column vector will be the outer product of the second column vector and the third column vector. T_{LV} is determined in this way because the unique transformation matrix between $\{\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L\}$ and $\{\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V\}$ must be an orthogonal matrix with a determinant 1. Consequently, we have constructed the transformation matrix T_{LV} between *LOBOTFrame* and *VehicleBodyFrame* from T_{AV}, T_{MV}, the accelerometer readings and the magnetic sensor readings, such that

$$(\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L) = (\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V) \cdot T_{LV} \tag{12}$$

All the above computation involves only a limited number of basic arithmetic operations. Considering that an orthogonal matrix has its inverse being its transpose, we have

$$\begin{aligned}(\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V) &= (\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L) \cdot T_{LV}^{-1} &(13)\\ &= (\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L) \cdot T_{LV}' &(14)\end{aligned}$$

Therefore, we have achieved expressing $\{\widehat{X}_V, \widehat{Y}_V, \widehat{Z}_V\}$ in terms of $\{\widehat{X}_L, \widehat{Y}_L, \widehat{Z}_L\}$ through limited algebraic arithmetic operations and thus determined the orientation of the vehicle. The question whether the robotic vehicle is moving forward or backward can be decided from the readings (positive or negative) of the rotation sensors.

Note that the above derivation assumes that the readings of the accelerometer reflect the gravitational force. When the robotic vehicle is moving, the accelerometer measurement often involves the movement acceleration. However, the movement acceleration for such a robotic vehicle is usually a very small fraction of the gravitational acceleration. As verified in our experiments, the effect of movement acceleration is negligible; even if it might show a considerable value during speeding up and braking, the time elapse in which it occurs is so short that it almost has no observable effect to localization.

2.3 Travel Distance

After inferring the instantaneous orientation of the robotic vehicle, we also need to know the momentary travel distance so as to compute the momentary relative movement. The rotation sensor attached to a motor continually measures the rotating angle. Let r be the rotation sensor reading in degrees, d be the wheel's diameter, then the travel distance of the wheel's movement is $\frac{r \cdot \pi \cdot d}{360}$. In the case of slippage and obstacle, a few recent research projects have been developed to handle such issues using methods such as sensing modalities and obstacle avoidance [18].

Another important issue we need to address relates to the way the robotic vehicle operates its motors. It is common that a robotic vehicle may make turns or follow a curved path through adjusting its two sides of motors at different speeds and even in reverse direction. Now, the question is how to calculate the moving distance given two different rotation sensor readings, one on each side. First, we observe that any small segment of movement, in a short enough time, can be perceived as part of a circular movement around a certain origin.

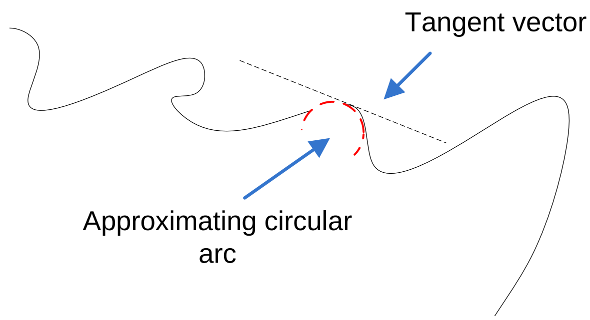
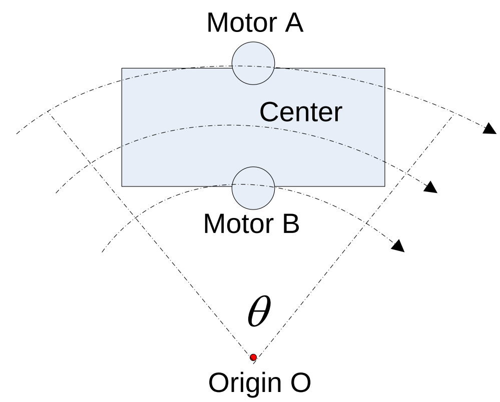


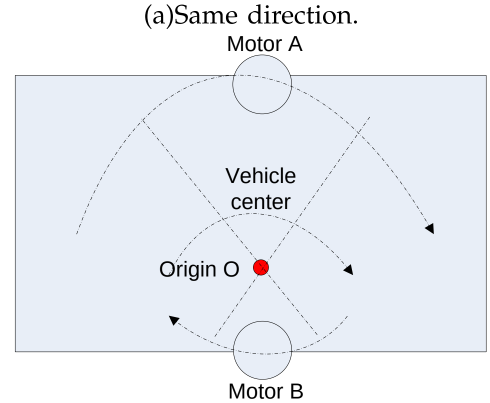
Fig. 3. Approximate curved path locally by circular arcs.

This observation can be made even when the two sides of wheels move in reverse direction. As an extreme scenario, when the vehicle makes a turn by reversing the two sides of motors at exactly the same magnitude of speed, the approximating arc has a radius of zero. In mathematical terms, a local curve, if short enough, can be approximated by a small arc with the same curvature and tangential at the intersection, as illustrated in Fig. 3. The curvature reflects how fast the curve turns at a point and depends on both the first derivative and second derivative of the curve. Approximating a curve locally with such an approximating arc produces a negligible cumulative difference when computing distance; that is because the approximating arc locally has almost the same first and second derivatives.

We claim that the travel distance of the robotic vehicle can be approximated by the average of the two side motor's travel distance. A motor may rotate either forward or backward; it rotates forward (backward) in an attempt to move the vehicle forward (backward). Correspondingly, in addition to the absolute distance measured, each reading of rotation sensor is assigned a sign: positive for forward rotation and negative for backward rotation. When the two sides' motors are moving in reverse direction, a positive distance is recorded as one side's reading and a negative distance for the other side. The robotic vehicle's direction is determined by the resulting average's sign. First, we discuss the case when the two motors are moving in the same direction but at different pace. As illustrated in Fig. 4(a), the center of the vehicle moves in an arc equally between Motor A's trace arc and Motor B's trace arc. It is straightforward that the center's arc length is the average of Motor A's arc length and Motor B's. Thus, we just theoretically proved the claim in the case that Motor A and B move in the same direction but at different pace. Next, we discuss the case that Motor A and B move in reverse direction. In this case, as shown in Fig. 4(b), the origin O around which the whole vehicle almost circularly moves is between the two motors. It is closer to the one with the smaller absolute pace. A bit straightforward geometry shows that the center's travel distance is the average of Motor A's and B's, with Motor A and B having different signs. The sign of the average determines the moving direction of the vehicle center.



(a)Same direction.



(b)Reverse direction.

Fig. 4. Travel distance with different-pace motors.

2.4 Drift Correction

As in many inertial systems, the localization computed through movement direction and travel distance tends to show drifting effect after a while. Fig. 5 compares the the trace retrieved in one of our outdoor experiments through our local relative positioning and through GPS. We observe that positioning purely through local relative positioning gradually drifts from the correct position and finally accumulates large error. Thus, LOBOT needs to apply drift correction to the localized results by utilizing the absolute position obtained from GPS.

LOBOT requests GPS sampling in an adaptive way that incorporates both location accuracy and energy use. The more frequent GPS sampling likely results in better correction of positioning; but more frequent GPS sampling also means significantly higher cost of power consumption [19], [20], [21]. Roughly, LOBOT adjusts its GPS sampling frequency according to the magnitude of the cumulative error of the local relative positioning. When the cumulative error of the local relative positioning between the current GPS sampling and its preceding GPS sampling increases, LOBOT increases its GPS sampling frequency accordingly; otherwise, LOBOT reduces its GPS sampling frequency. Specifically, let $CErrThd$ be the tolerant threshold of the cumulative error of

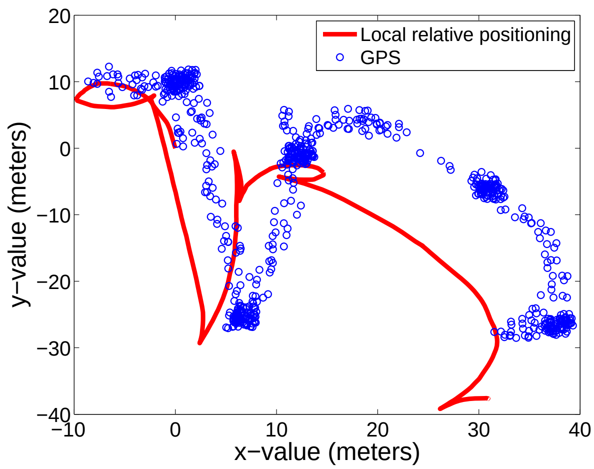


Fig. 5. Drift in local relative positioning.

the local relative positioning between two consecutive GPS samplings; P be the time elapse between the two most recent GPS samplings; $CErr$ be the cumulative error of the local relative positioning between these two most recent GPS samplings. Then the time elapse from the most recent GPS sampling to the next GPS sampling will be $P \cdot CErrThd/CErr$. $CErr$ is calculated on the runtime as the difference between the current GPS reading and the location computed purely based on the local relative positioning and the previous GPS reading. In our outdoor experiments, we used 8 meters as the value of $CErrThd$. Initially, a value is set up as the gap between the two first GPS samplings; then, the next gap will be calculated as $P \cdot CErrThd/CErr$; after that, a third gap can be calculated in the same way based on the second gap. Thus, we only need to set up the initial gap value and the subsequent time elapses will evolve by themselves. In our outdoor experiments, we used 2 minutes as the initial gap. In practice, to increase stability, LOBOT adopts a GPS sampling gap period slightly lower than $P \cdot CErrThd/CErr$. When the GPS signal is not available, LOBOT periodically wakes up the GPS receiver to check its availability and then puts it to sleep.

LOBOT assumes identical distribution of cumulative error among all time periods of equal length. Let the probability sample space be the set X of all possible localization-related events, $err(X,t)$ be the random error of local relative positioning at time t, and $corr(X,t)$ be the correction at time t. $corr(X,t)$ is the difference between the position obtained through relative positioning and the ground truth. err and $corr$ are both stochastic processes. Let the time start at 0 (last successful GPS request), end at T(the current GPS reading time); assume LOBOT performs local relative positioning at time $1, 2, 3..., T-1, T$. Here we analyze the correction with these simplified assumptions in mind; in fact, our reasoning works with a more general situation with the same logic. Then $corr(X,0) = 0$. We have

$$corr(X,t) = \sum_{i=0}^{t} err(X,i), 0 < t < T \tag{15}$$

According to the maximal-likelihood estimation, an optimal estimate of $corr(X,t)$ is its mean value

$$\begin{aligned} E(corr(X,t)) &= \sum_{i=0}^{t} E(err(X,i)) & (16) \\ &= t \cdot E(err(X,1)) & (17) \end{aligned}$$

We also have

$$E(err(X,1)) = E(corr(X,T))/T \tag{18}$$

Therefore, combining the above two equations, we have

$$\begin{aligned} E(corr(X,t)) &= t \cdot E(err(X,1)) & (19) \\ &= t \cdot E(corr(X,T))/T & (20) \end{aligned}$$

Again, based on the principle of maximal-likelihood estimation, the mean value $E(corr(X,T))$ has its estimated value being the difference between the current GPS-supplied reading and the last position obtained through relative positioning. Additionally, an optimal estimate of the random correction $corr(X,t)$ at time t is $t \cdot E(corr(X,T))/T$. Therefore, to correct the drift at time t, we only need to estimate $E(corr(X,T))$ and then add $t \cdot E(corr(X,T))/T$ to the original position estimate. $E(corr(X,T))$ is estimated to be the difference between the current GPS-supplied reading and the last position obtained through local relative positioning.

Finally, it is possible that LOBOT is inactivate first and then becomes active when there is no GPS signal. In this situation, LOBOT is only able to compute its relative movement until it receives a GPS signal in the future. Once a GPS sampling is available, it starts to trace back and restore all the absolute location before that point. If no GPS signal is available, LOBOT will interpolate one of it absolute position linearly with respect to time and derive the rest using its recorded relative movement.

3 Implementation And Empirical Evaluation

To implement LOBOT, we used a low-cost LEGO MINDSTORM NXT 2.0 vehicle robot [22] and a moderately priced HTC Legend smart phone [23] as shown in Fig. 6. The HTC Legend phone is mounted onto the robot, merely to supply a set of sensors: an accelerometer, a magnetic sensor and a GPS. In our experiments, the HTC phone is lifted higher to avoid the magnetic interference from both the robot and the ground. Powered by six AA batteries, this LEGO NXT robot moves on its two servo motors (one on the left and the other on right). The two servo motors can rotate at their own user-specified speeds, either in the same direction or reverse, providing flexible movement. Their rotating speeds can be changed by user programs at any moment. The LEGO NXT has a

Fig. 6. The LEGO NXT robot and the HTC Legend phone.

set of built-in rotation sensors to continually measure the rotating distance of each motor. The HTC Legend phone has an accelerometer (G-sensor), a magnetic sensor (digital compass) and an internal GPS. Our programs control the motor's movement, collect the data from rotation sensors, the accelerometer, the magnetic sensor as well as GPS.

We performed repeated experiments indoors and outdoors on the main campus of Wayne State University, scaling from 1m x 1m (meter) areas up to areas of 50m x 50m. The LEGO robot randomly moves from its minimal speed (the speed of a snail) to its full speed (several inches per second) and may change its speed and direction every few seconds. It may also operate its two motors at different pace or reversely to follow curved path and make turns. These experiments computed the location data on all three axes: x (East), y (North) and z (upward). Each experiment lasts from 1 minute to 20 minutes. The programmed robot randomly decided its next movement after every certain amount of time from 5 seconds to 1 minute.

The two approaches, LOBOT and the purely accelerometer-based approach, were both executed simultaneously during each experiment. The GPS raw data were collected during outdoor experiments when applicable. To get the ground truth, we performed manual recording of positions in most cases and camera-assisted positioning in small areas. Our experiments indicate that the purely accelerometer-based approach can not achieve satisfactory results within the context of localizing a ground robotic vehicle like the LEGO robot we used. In contrast, LOBOT, with a low-cost setting, realizes relatively accurate positioning either indoors or outdoors. Although the pure local relative positioning component of LOBOT shows the cumulative drifting effect, LOBOT well compensates the drift through the infrequent GPS-augmentation.

3.1 Inaccuracy of Sensing Data

Before dipping into the detailed performance analysis, we would like to observe the inaccuracy of the received sensing data. The sensing data usually display certain deviation from the true sensing value due to various issues from the hardware or the software. When such inaccuracy starts to accumulate, the resulting location might noticeably deviate from the ground truth. A successful localization system should at least be able to reduce the cumulative errors. It is noteworthy that the various positioning techniques often differ not by their theoretical soundness, but by their capability to resist data inaccuracy. The purely accelerometer-based positioning approach has its strong theoretical foundation from the Newton's Second Law of Motion; however, the position resolved from the acceleration data might quickly deviate from the ground truth. Admittedly, our LOBOT system is also impacted by the cumulative error from the rotation sensor, the accelerometer, the magnetic sensor and the GPS. Fortunately, in the first place, LOBOT tends to have much lower cumulative error than the accelerometer-based approach; further, after performing the GPS-augmentation, the remnant of the cumulative error is well under an acceptable range, considering the low cost of LOBOT.

While these sensors are capable of capturing instantaneous movement, the accuracy of the positioning results are strongly impacted by the specific localization approaches being used. The sensing error varies, depending on the sensors. Generally, the magnetic sensor, motor rotation sensor, and the accelerometer tend to show small instantaneous sensing error; the GPS receiver may produce a relative large error in location. The very small instantaneous inaccuracy of the acceleration data could lead to large positioning errors if the acceleration is used as the exclusive raw data for positioning. That is due to the major quadratic effect in computing the travel distance from the acceleration: $S = vt + \frac{1}{2}at^2$, a being the acceleration. Even with a perfect instantaneous acceleration, the inaccuracy resulting from applying that value as estimation for a whole small time interval could be detrimental.

While the purely acceleration-based schemes may suffer from the quadratic effect, LOBOT involves only linear computation among the raw data. It tends to accumulate errors much slower than the accelerometer-based approach. Although a single GPS reading can have error of up to three meters in our experiments, unlike the relative position based on accumulation, the GPS positioning does not accumulate errors: a previous inaccurate GPS reading would not affect the current GPS reading. Finally, when the GPS-augmentation is applied to the drifting outcome of the local relative positioning component, the resulting location solution is satisfactory.

3.2 Evaluation of Local Relative Positioning

LOBOT strongly relies on the low cumulative errors of its local relative positioning component. A major portion of the experiments were performed to evaluate the local relative positioning. Both the manual measurement and the camera-assisted positioning were used to gain the ground truth. Though most results are from experiments on relatively flat planes (2D experiments), we also carried out 3D experiments of localizing the robot on surfaces with a slope. LOBOT does not favor one dimension over another. As a matter of fact, any two dimensions from a 3D experiment can be viewed as a 2D experiment. For that reason, the major analysis is on the 2D experiments while the 3D experiments exhibit similar characteristics.

3.2.1 Two-Dimensional Experiments

We present the 2D trace of the robot as well as the time series of the movement on each single dimension. The results show the relatively low cumulative errors of LOBOT and the large deviation of the purely accelerometer-based approach.

According to our 10 experiments with each running 20 minutes in 12m x 12m areas, the trace resulting from LOBOT has an accuracy of within 2.5 meters compared to manual recordings. One such experiment is shown in Fig. 7. In Fig. 7, the (x, y) coordinates by LOBOT are relatively close to the manual recordings. In contrast, the accelerometer-based approach tends to suggest almost "no-movement" on the plane and dramatic movement on the third dimension (the altitude). As in Fig. 7, the results from the accelerometer-based approach falsely "suggest" that the robot moves within a small circle with 1m radius. Since the movement is on flat plane surfaces, LOBOT naturally verifies the limited movement on the third dimension. The altitude from LOBOT is within a range from -0.5m to 0.5m through 20 minutes. One such example is presented in Fig. 8. In contrast, the accelerometer-based approach often falsely reports a dramatic movement on the third dimension. Again as in Fig. 8, according to that approach, the robot is driving down a steep slope though it never leaves the flat plane ground. As for such results, it is reasonable to suspect that the acceleration data on the third dimension might have a constant large negative deviation from its true zero value and that the deviation could have resulted from an inaccurate gravitational constant or simply the sensing errors. However, the acceleration data on the third dimension seems to suggest only very small constant deviation of the acceleration data might exist. The corresponding data for the same previous experiment is extracted and shown in Fig. 9. The figure indicates that the acceleration data oscillates around zero. To explain the dramatic error on the z-value of the accelerometer-based approach, we note that this approach involves a quadratic expression of the time and thus the time elapse accumulates such errors very fast.

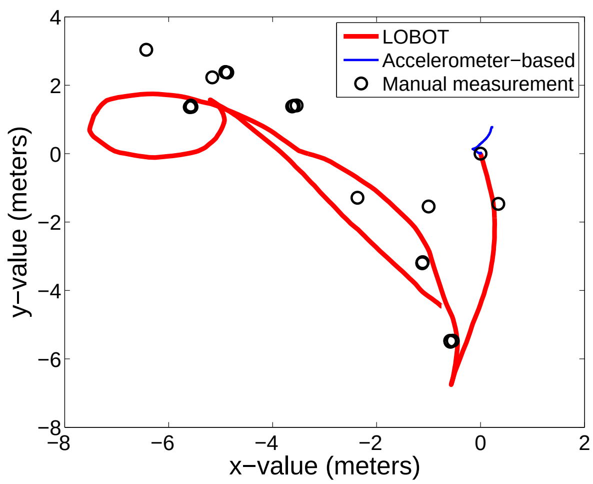


Fig. 7. Trace comparison of a 2D experiment.

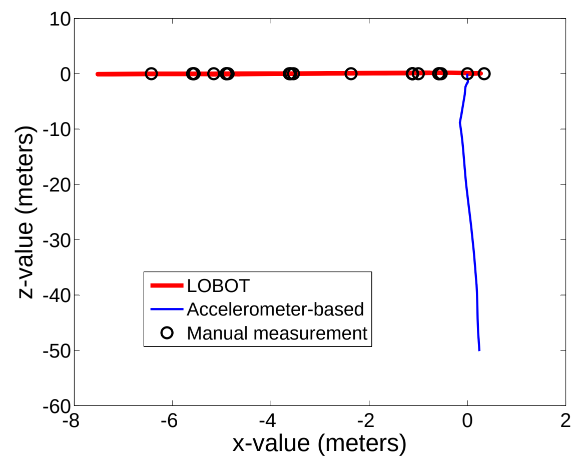


Fig. 8. (x,z) trace comparison.

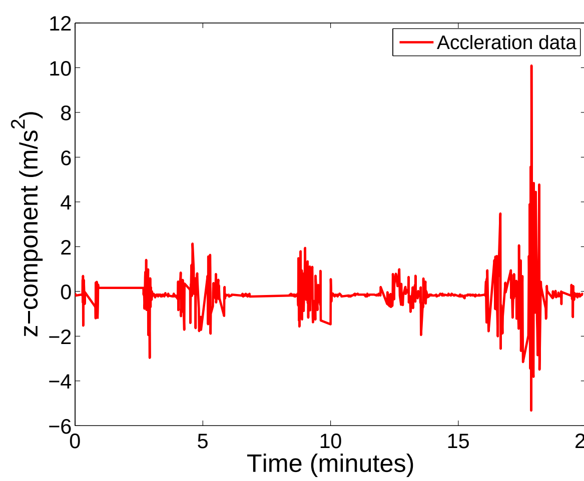


Fig. 9. Sensing data: acceleration

In addition to the trace, the time series of the components of the movement vector on each dimension also confirms the satisfactory performance of LOBOT's local relative positioning. With the same experiment in Fig. 7, the time series of the x is almost perfectly close to the ground truth. The time series of y values is plotted in Fig. 10. The y values of LOBOT exhibit a deviation of up to 1.75m over 20 minutes. On the other hand, as for the accelerometer-based approach, the figure displays almost static y values.

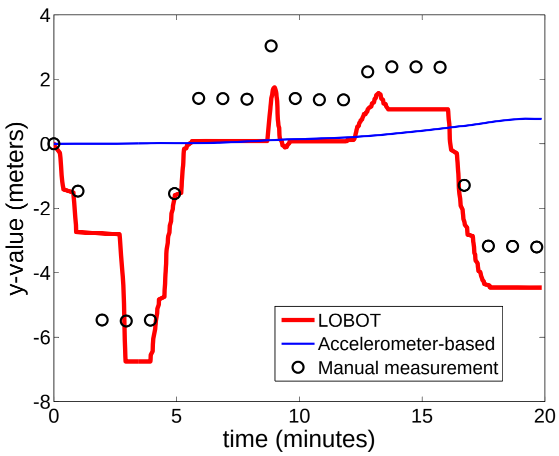


Fig. 10. Time series of y-value

Finally, getting the ground truth through the camera-assisted positioning allows better examination of LOBOT. As found in our experiments, the error of LOBOT generally accumulates slowly; however, occasionally a relative noticeable transient error occurs due to accidents such as slippage. Despite the cumulative errors, the trace LOBOT retrieves generally follows the overall movement trend. As in one experiment (Fig. 11), the robot moved for one minute, over which the local relative positioning performs almost perfectly except when a slippage occurred around the position (-0.07, 0.33). After the slippage, the trace curve still has a very similar shape as the camera-retrieved ground truth, however, with a shifting effect. When such a noticeable error happens, after the GPS-augmentation, the results can often be adjusted to be relatively close to the ground truth.

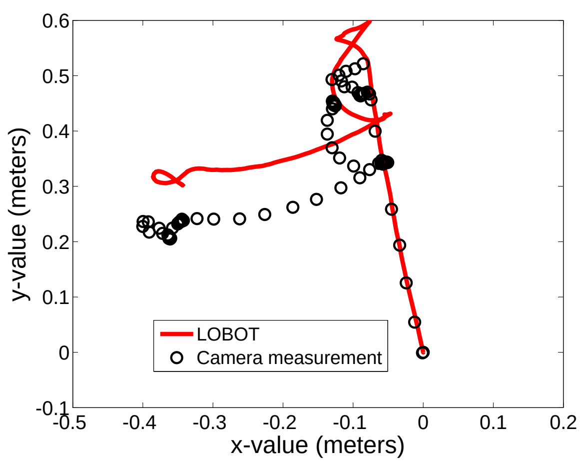


Fig. 11. LOBOT trace with cumulative errors.

3.3 Evaluation of LOBOT with GPS-Augmentation

We performed a few outdoor experiments in GPS-available areas of up to 50m x 50m. To obtain the ground truth, the GPS on the HTC Legend phone is turned on and computes positions at least once every three seconds. Since the GPS's (longitude, latitude) data can be locally viewed as Cartesian coordinates, we mapped the GPS data onto a meter-based distance coordinate through linear regression. The trace produced by LOBOT is compared against the continuous GPS timestamped trace. The empirical analysis shows that the LOBOT's local relative positioning produces an inaccuracy of up to 18m; with one-time GPS-augmentation, the error is well under 8m. Without the GPS-augmentation, the trace retrieved still has a similar shape to the ground truth but with a drift. The result of one experiment is illustrated in Fig. 12. In Fig. 12, the thicker red line is the trace produced by the LOBOT without the GPS-augmentation, the small circles are the GPS trace, and the thinner green line is the trace by LOBOT with the correction from the last GPS-detected position. The one-time adjustment from the GPS data largely corrects the drift. With the

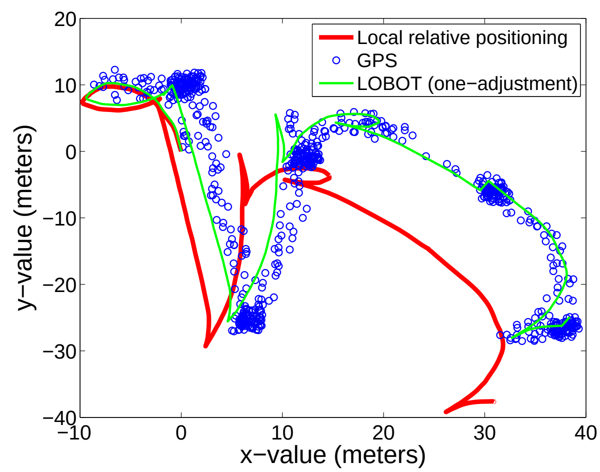


Fig. 12. Outdoor experiments with one-time GPS-augmentation.

same experiment, we performed a two-time adjustment: first correction based on the GPS data collected in the middle of the experiment time; the other correction based on the last GPS data. Interestingly, the two-time adjustment does not seem to suggest much improvement over the one-time adjustment, as shown in Fig. 13. The main reason is, the GPS measurement itself is known to have inherent inaccuracy.

4 Conclusions

We propose LOBOT, a low-cost, self-contained, accurate localization system for small-sized ground robotic vehicles. LOBOT localizes a robotic vehicle with a hybrid approach consisting of infrequent absolute positioning through a GPS receiver and local relative positioning based on a 3D accelerometer, a magnetic field sensor and several motor rotation sensors. LOBOT fuses the information from an accelerometer, a magnetic sensor and motor rotation sensors to infer the movement of

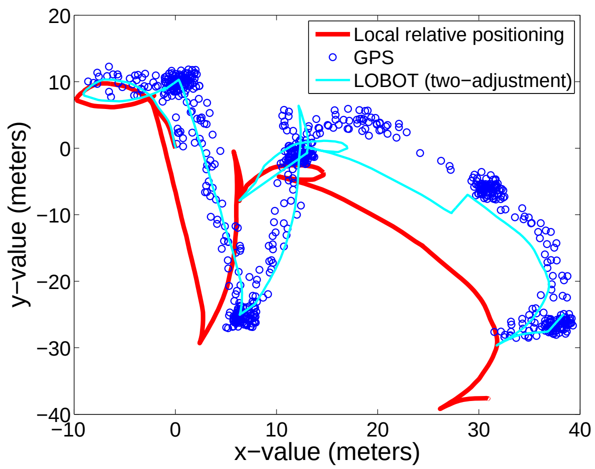


Fig. 13. Outdoor experiments with two-time GPS-augmentation.

the robot through a short time period; then the inferred movement is corrected with infrequent GPS-augmentation. The hardware devices LOBOT uses are easily-available at low cost. LOBOT is self-contained in that it virtually requires no external devices or external facility management and that it needs no prior information. Unlike other localization schemes such as radio-based solutions, LOBOT does not require external reference facilities, expensive hardware, careful tuning or strict calibration. Additionally, LOBOT applies to both indoor and outdoor environments and realizes satisfactory performance. We developed a prototype of LOBOT and conducted extensive field experiments. The empirical experiments of various temporal and spatial scales with LOBOT verified its accuracy. In contrast to the accelerometer-based approach, LOBOT succeeds in maintaining low cumulative error. The GPS-augmentation greatly enhances LOBOT's resilience.

References

[1] K. Yu and I. Oppermann, "Uwb positioning for wireless embedded networks," in *Radio and Wireless Conference, 2004 IEEE*, sept. 2004, pp. 459 – 462.

[2] L. Thiem, B. Riemer, M. Witzke, and T. Luckenbach, "Rfid-based localization in heterogeneous mesh networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 415–416. [Online]. Available: http://doi.acm.org/10.1145/1460412.1460480

[3] N. Petrellis, N. Konofaos, and G. Alexiou, "Target localization utilizing the success rate in infrared pattern recognition," *Sensors Journal, IEEE*, vol. 6, no. 5, pp. 1355 –1364, oct. 2006.

[4] J. Hesch, F. Mirzaei, G. Mariottini, and S. Roumeliotis, "A laser-aided inertial navigation system (l-ins) for human localization in unknown indoor environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 5376 –5382.

[5] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp, "Walrus: wireless acoustic location with room-level resolution using ultrasound," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, ser. MobiSys '05. New York, NY, USA: ACM, 2005, pp. 191–203. [Online]. Available: http://doi.acm.org/10.1145/1067170.1067191

[6] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The effects of ranging noise on multihop localization: an empirical study," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, april 2005, pp. 73 – 80.

[7] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, pp. 41–52, January 2007. [Online]. Available: http://doi.acm.org/10.1145/1234822.1234829

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom'00)*, Aug. 2000.

[9] X. Cheng, A. Teler, G. Xue, and D. Chen, "Tps: a time-based positioning scheme for outdoor wireless sensor networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, march 2004.

[10] J. Liu, Y. Zhang, and F. Zhao, "Robust distributed node localization with error management," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 250–261. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132933

[11] M. Maróti, P. Völgyesi, S. Dóra, B. Kusý, A. Nádas, A. Lédeczi, G. Balogh, and K. Molnár, "Radio interferometric geolocation," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ser. SenSys '05. New York, NY, USA: ACM, 2005, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1098918.1098920

[12] G. Desouza and A. Kak, "Vision for mobile robot navigation: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237 –267, feb 2002.

[13] P. Lamon and R. Siegwart, "Inertial and 3d-odometry fusion in rough terrain - towards real 3d navigation," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, sept.-2 oct. 2004, pp. 1716 – 1721.

[14] B. Liu, M. Adams, and J. Ibanez-Guzman, "Minima controlled recursive averaging noise reduction for multi-aided inertial navigation of ground vehicles," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, aug. 2005, pp. 3408 – 3414.

[15] N. Schmitz, J. Koch, M. Proetzsch, and K. Berns, "Fault-tolerant 3d localization for outdoor vehicles," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, oct. 2006, pp. 941 –946.

[16] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar, "Autowitness: locating and tracking stolen property while tolerating gps and radio outages," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 29–42. [Online]. Available: http://doi.acm.org/10.1145/1869983.1869988

[17] J. Kim, J. Lee, G. Jee, and T. Sung, "Compensation of gyroscope errors and gps/dr integration," in *Position Location and Navigation Symposium, 1996., IEEE 1996*, apr 1996, pp. 464 –470.

[18] G. Reina, L. Ojeda, A. Milella, and J. Borenstein, "Wheel slippage and sinkage detection for planetary rovers," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 2, pp. 185 –195, april 2006.

[19] N. Priyantha, D. Lymberopoulos, and J. Liu, "EERS: Energy efficient responsive sleeping on mobile phones," in *Proceedings of PhoneSense 2010*, Nov. 2010.

[20] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 299–314. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814463

[21] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 315–330. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814464

[22] "Lego mindstorm nxt 2.0," http://mindstorms.lego.com, 2009.

[23] "Htc legend," http://www.htc.com/europe/specification.aspx?p_id=313, 2010.

Guoxing Zhan Guoxing Zhan expects to receive his Ph.D. degree in August 2012 in Computer Science from Wayne State University. He received a M.S. in Mathematics from Chinese Academy of Sciences in 2007 and another M.S in Computer Science from Wayne State University in 2009. Mr. Zhan is interested in research on participatory sensing, wireless sensor network, mobile computing, networking and systems security, trust Management, and information processing. Several of his research papers have been presented at international conferences or published in journals. Additionally, Mr. Zhan has instructed a few computer science labs consisting of interactive short lectures and hands-on experience.

Weisong Shi Dr. Weisong Shi is an Associate Professor of Computer Science at Wayne State University. He received his B.S. from Xidian University in 1995, and Ph.D. degree from the Chinese Academy of Sciences in 2000, both in Computer Engineering. His current research focuses on computer systems, mobile and cloud computing. Dr. Shi has published more than 100 peer reviewed journal and conference papers. He is the author of the book "Performance Optimization of Software Distributed Shared Memory Systems" (High Education Press, 2004). He has served the program chairs and technical program committee members of several international conferences. He is a recipient of the NSF CAREER award, one of 100 outstanding Ph.D. dissertations (China) in 2002, Career Development Chair Award of Wayne State University in 2009, and the "Best Paper Award" of ICWE'04 and IPDPS'05.