

# Low-cost and Accurate 3D Road Modeling Using Mobile Phone

Zhice Yang, *Student Member, IEEE* Qian Zhang, *Fellow, IEEE*

**Abstract**—This paper studies the feasibility of using mobile phone to achieve low-cost and accurate 3-dimensional (3D) road modeling, which provides nutritious supplement to current digital map. Instead of using expensive tools, our method only needs a mobile phone fixed on a moving vehicle. Road surface variation, reflected by vehicle dynamic state, is captured by sensors in the mobile phone. The road in 3D is reproduced by combining sensing data from accelerometer, magnetometer, gyroscope and GPS trace. Several compensation and calibration techniques are developed to improve modeling accuracy based on the systematic study of sensor error properties. A learning step is used to automatically unify the user dependent factors to avoid manual adjustment. We evaluate our method on a 2 km real road. Compared with the field measured baseline, it successfully generates a fine-grained 3D road model with maximum height error less than 4m and 90% slope angle error less than 2 degrees.

**Index Terms**—Mobile phone, Measurement, 3D Road Map, Sensor

## 1 INTRODUCTION

THE rapid spread of mobile phone navigation systems and the fast development of wireless network have dramatically popularized the 2-dimensional (2D) digital map. Due to the dimensional gap between 2D map and the real 3D world, 2D digitalization will never be the ultimate goal of the map evolution. Consider a 3D map providing detailed road slope angle information. It will be extremely helpful for drivers encountering bad weather such as snow and sleet. Using the maximum inclining angle on each road segment, intelligent driving system can automatically route to avoid roads on which vehicles are likely to slip. Such accurate 3D road map is also meaningful in improving driving safety [1], reducing energy consumption [2] and enabling new vehicle localization approach [3].

Despite numerous potential benefits, accurate 3D road map is not easy to generate. The key reason is the number of roads in the world is huge and keeps growing. Owing to this fact, there is a trade-off between the accuracy and the expense in traditional methods. Specifically, large-scale measurement methods such as satellite remote sensing and aerial photographing [4] cannot provide images with enough resolution to extract road surface variation. These methods also suffer from unexpected barriers such as trees beside mountainous roads. To avoid those drawbacks, other approaches equip helicopters or vehicles with special

devices such as laser scanners [5], sensors [1] and cameras [6] to enable 3D road measurement. Though accurate, the costly instruments, dedicated labor and a great deal of time become the major concerns for nationwide deployment.

In this paper, we try to seek a method to keep measurement burden minor while achieve enough accuracy to fulfill emerging demand. The exploring hints come with two observations. First, modern commodity mobile phone is programmable and equipped with various Microelectromechanical Systems (MEMS) sensors, which make the mobile phone an ubiquitous sensing platform. Second, the road surface variation is directly related to the vehicle spatial dynamics. In other words, car tilts up and down when encounters slopes. The above two observations show the insight that the mobile phone could be a 3D road measurement tool. Millions of drivers carrying mobile phones are not encouraged to make phone call during driving. In this sense, mobile phones in moving vehicles, once fixed, are able to monitor roads variation without affecting their normal use. Moreover, such cooperative measurement stretches spatial coverage with huge user amount and car mobility. Therefore, the measurement cost is shared by undedicated participants.

The simple intuition incurs a main concern: Is it possible to get 3D road information *accurately* and *easily* through mobile phone only? We answer it by proposing our method *3Rd*. *3Rd* consists of mobile phone clients and a backend server. The client app works when driver fixes his/her mobile phone on the vehicle, and mainly performs sensing and recording. The only constraint is that the mobile phone should be fixed to the vehicle during the measurement. We find

• Z. Yang and Q. Zhang are with Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong SAR, China.

the phone holder <sup>1</sup> is a common accessory to ease this work. Besides this, priori to first use, the driver needs to finish a closed loop path to generate a profile for that vehicle to aid measurement. The backend server is responsible for calculating 3D road model based on the sensing data from clients. The key principle is that accelerometer reading of the constant gravity direction differs on roads with different slope angles. To accomplish it, we fuse the data from accelerometer, magnetometer and gyroscope to reproduce the vehicle position sequence in 3D space. The position sequence is then used to generate moving directions of that vehicle. Further, moving directions are treated as the slope angles and sticked to GPS trace to obtain the 3D road model.

*3Rd* achieves accurate measurement while keeps concise procedures by addressing two practical challenges. First, low-cost MEMS sensors on commodity mobile phone are coarse and inaccurate [7]. The design of *3Rd* is based on the knowledge of the error properties of each sensor, and incorporates errors and uncertainties by comprehensively formulating the measurement problem. For instance, GPS trajectories are used to reduce acceleration shaking yet without introducing new measurement error. Gyroscope measurement with bias error is high-pass-filtered to provide fine-grained slope variation. More techniques are developed and combined in our design to target this challenge; we put their details in the following sections.

The second challenge is owing to the fact that mobile phone is a personal device. It suffers from non-uniform user behaviors and different environmental factors. For example, different users may have their phone holders placed at different positions, and thus the position relation between the mobile phone and the vehicle is unknown. Those user-depend factors, though inevitable, should not be manually adjusted. In *3Rd*, we introduce a learning step to construct the personal measurement profile. *3Rd* intentionally runs a closed loop path for reference and leverages the unique path features to estimate the phone holder position and the in-car electromagnetic fields to guide the real measurement. All of the user-dependent parameters are determined automatically in the learning step. Users do not require to provide further adjustment until next structure resetting such as replacing a car or changing the phone holder direction.

Contributions of this work are:

- As far as we know, this is the first work that studies the feasibility of using mobile phone to model roads in 3D space. The low-cost nature makes large scale 3D road generation possible and its high accuracy could benefit new applications.
- We design and implement a 3D road modeling

method based on commodity mobile phone sensing. Various techniques are created to increase modeling accuracy and improve usability.

- We evaluate our method in a 2km real road. Results show it has a comparable accuracy to the field measurement. *3Rd* successfully produces a fine-grained 3D road model with maximum height error less than 4m and 90% slope error less than 2 degrees.

The rest of this paper is organized as follows. §2 gives an overview the modeling method. §3 formulates the problem and describes the algorithm in detail. §4 introduces the implementation. The evaluation is presented in §5. We discuss an application example and other related issues in §6. The related work is in §7. Finally, we conclude our work is in §8.

## 2 DESIGN OVERVIEW

In this section, we first show the intuition of inferring road shape based on measurable parameters. Then we give an overview of our design.

### 2.1 Measurement Basis

Regardless of the complex 3D vector mathematical calculation, our measurement basis is quite simple and intuitive. One obvious fact is that vehicle dynamic states, such as running up or down, turning left or right, are determined by the shape of the road. Sensors in the mobile phone, which is fixed to the vehicle, are able to detect these state changes.

In Fig.1, a car carrying a mobile phone is moving from the flat side to the uphill. Frame  $O^b$  represents the mobile phone body and frame  $O^n$  represents the Earth/map.  $O^b$  twists according to the slope while  $O^n$  is stationary. The moving direction vector, whose direction is parallel to the road surface and points to the front of the car, has two groups of readings when viewed in different frames. In  $O^n$ , the moving direction vector points to the tangent direction of the road, and can be used to derive road angles. In  $O^b$ , the vector is a constant and can be predetermined. In order to determine the moving direction in  $O^n$ , *3Rd* first uses the natural constant vectors, the gravity and the magnetic North, as reference to calculate frame  $O^b$  variations through sensor readings. Then, *3Rd* transforms the known moving direction vector in  $O^b$  into  $O^n$  to obtain the slope angle sequence. Road model in 3D is finally generated by sticking the slope angle sequence to the GPS trace.

### 2.2 Modeling Steps

*3Rd* takes three regular steps to model the road and one additional learning step before first use. All of the calculations are performed in the server and clients are only responsible for sensing and collecting data.

*Measurement initialization and data collection.* We fix the mobile phone on the vehicle to monitor its position changing during the measurement. With this

1. [http://www.amazon.com/s/ref=nb\\_sb\\_noss?url=search-alias%3Dmobile&field-keywords=Phone+Holder+](http://www.amazon.com/s/ref=nb_sb_noss?url=search-alias%3Dmobile&field-keywords=Phone+Holder+)

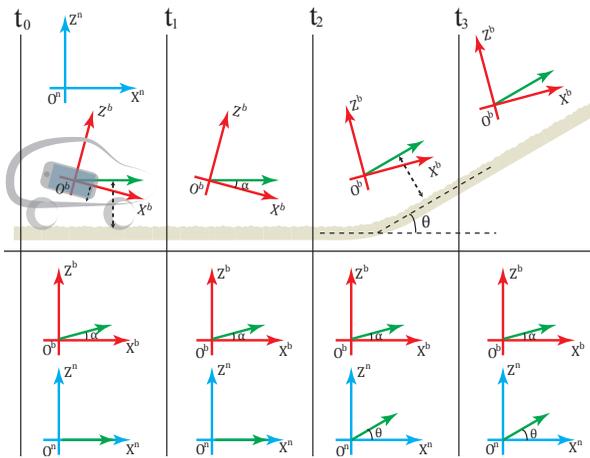


Fig. 1. Measurement Basis Illustration

condition, the moving direction vector becomes a constant with respect to the mobile phone body  $O^b$ , and it can be measured as profile with in the *Closed Loop Learning* step.

*Frame transformation calculation.* This step uses sensing data to calculate the 3D position of the mobile phone body with respect to the Earth. The position is useful in transforming the reading of the moving direction with respect to the phone body  $O^b$  to the reading with respect to the Earth  $O^n$ . The calculation is done in two ways. One method uses the gravity direction and the magnetic North direction to calculate the mobile phone position. Another method starts from an initial position, which is propagated with incremental rotation values recorded by gyroscope. Due to the different error properties, we implement both and fuse them in the *3D road modeling* step.

*3D road modeling.* *3Rd* models the road in terms of height and angle. Here, the moving direction vector in  $O^b$  is known as user profile. First, the slope angle sequence is calculated by transforming the moving direction from  $O^b$  into  $O^n$ . Then, the angle sequence is stucked to GPS trace according to the time stamps. After that, we enhance the slope angle accuracy by exploiting the error properties of the two transformation methods. Finally, the relative height of the road is obtained by adding the products of the tangent of slope angle and the corresponding horizontal displacement.

*Closed Loop Learning.* The learning step is a must to determine user profile before the first use. This step depends on the three steps above and solves unknowns with special input. It takes a closed loop path in the modeling process and searches the unknown parameters (moving direction with respect to  $O^b$  and local magnetic field) according to the the property of the a closed loop, which is a road path with the same start and end point. The loop path can be but not limited to a circle; driving around a block is a choice. Further, the learning is also automatically done when drivers encounter closed loop during in the regular

measurement.

### 3 DESIGN DETAILS

In this section, We first formulate the modeling problem and define the accuracy metrics. Then, we describe the method details of *3Rd*. We use the same subsection names to show the relation to the brief steps in §2.2.

#### 3.1 Notations and Problem Formulation

In this subsection, we give the concept of frame, formulate the modeling problem and define the measurement accuracy metrics.

##### 3.1.1 Frame

Frame or frame of reference is important because it provides physical meaning for the sensing data collected by mobile phones.

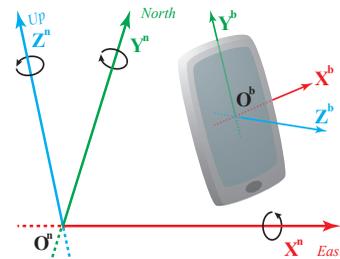


Fig. 2. b-frame and n-frame

The mobile phone body reference frame (b-frame  $O^b$ ) is fixed on the mobile phone body. A Cartesian coordinate system is aligned to this frame as Fig.2 shows. Its coordinate system and vector readings are noted with superscript  $b$ .

The navigation reference frame<sup>2</sup> (n-frame  $O^n$ ) is fixed on the Earth, whose coordinate system has its origin fixed at a selected point and axes aligned with the directions of the East, the North and the vertical (up). Its coordinate system and vector readings are noted with superscript  $n$ .

The reading of a vector in one frame is defined as its decomposition values along the frame coordinate axes. For example, the reading of a vector  $\vec{v}$  in b-frame is expressed as  $\vec{v}^b = [v_x^b, v_y^b, v_z^b]^T$ , which is defined as  $\vec{v} = v_x^b \vec{e}_x^b + v_y^b \vec{e}_y^b + v_z^b \vec{e}_z^b$ ,  $\vec{e}_x^b, \vec{e}_y^b, \vec{e}_z^b$  are the basis vectors of the corresponding b-frame coordinate system. Similarly, its reading in n-frame is  $\vec{v}^n = [v_x^n, v_y^n, v_z^n]^T$ .

The norm of a vector  $\vec{v}$  is its strength, denoted by  $|\vec{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2}$ . We use an orthogonal matrix  $C_b^n$  to denote the reading transformation between two frames:  $C_b^n \cdot \vec{v}^b = \vec{v}^n$   $C_n^b \cdot \vec{v}^n = (C_b^n)^T \cdot \vec{v}^n = \vec{v}^b$ .

The measurement is taken in discrete time, so we introduce  $t$  as the sampling time. The attribute  $\vec{v}$  is expressed in terms of  $t$  as  $\vec{v}(t)$ . If  $\vec{v}(t)$  is a constant, we ignore its variable as  $\vec{v}(t) = \vec{v}$ .

2. In spite of the earth rotation, in our measurement, we treat n-frame as an inertial frame, where Newton's law is valid.

### 3.1.2 Problem Formulation

A road on the Earth can be treated as a curve in n-frame. Fig.3 shows that a road curve is described by triples  $\vec{c}^n(t) = [c_x^n(t), c_y^n(t), c_z^n(t)]^T$  sequence in n-frame. Modeling the road in 3D space is to determine the value of  $\vec{c}^n(t)$  at every  $t$ . Without confusion, we ignore the superscript of  $\vec{c}^n(t)$  as  $\vec{c}(t)$  in the context.

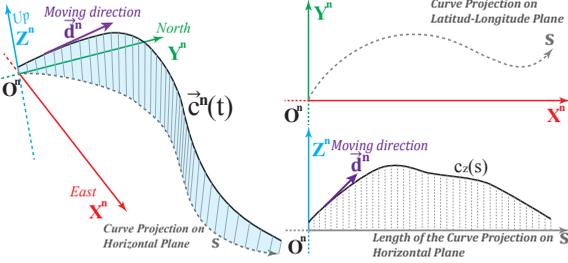


Fig. 3. Problem Formulation

The measurement problem consists of two parts. First, the horizontal locations  $c_x(t)$  and  $c_y(t)$ , shown in the dash line  $s$  in Fig.3, are the projection of  $\vec{c}(t)$  on the horizontal plane and can be measured by GPS longitude and latitude. The second part is the vertical dimension  $c_z(t)$ . Traditional 2D maps are lack of the information of  $c_z(t)$ , which is what we are going to measure. Note that  $s(t)$  is uniquely determined by  $t$ . Therefore,  $c_z$  can be expressed in terms of the horizontal distance:

$$c_z(t) = c_z(t(s)) = c_z(s) \quad (1)$$

The variation rate of  $c_z(s)$  is relates to the road slope angle  $\theta(s)$ . Note that  $\theta(s)$  is just the angle between the car moving direction  $\vec{d}^n(s) = [d_x^n(s), d_y^n(s), d_z^n(s)]^T$  and the horizontal plane:

$$c_z(s) - c_z(s_0) = \int_{s_0}^s c'_z(s_i) ds_i = \int_{s_0}^s \tan(\theta(s_i)) ds_i \quad (2)$$

$$\tan \theta(s) = d_z^n(s) / \sqrt{d_x^n^2(s) + d_y^n^2(s)} \quad (3)$$

Equation (2) and (3) indicate that we can obtain the relative road height by measuring the car moving direction along the road.

### 3.1.3 Accuracy Definition

Accuracy is the most essential part for a modeling method. We define two metrics here on purpose of fully describing the road in 3D space. We especially focus on the vertical dimension.

First, the geometric difference between the road model and the real road must be small. Taking the sampling points at  $s_1 \dots s_n$ , the modeling results are  $\vec{c}_m(s_i)$ . The maximum height error should be smaller than the required threshold:

$$\max_{i \in \{1 \dots n\}} |c_{mz}(s_i) - c_z(s_i)| \leq \alpha \quad (4)$$

Height difference, though intuitive, could be confusing. Take upper part of Fig.4 for example, the dash

line is the real road and the solid line is the modeling results. Three measurements have the same maximum error as  $1m$ .  $\alpha = 1m$  seems a perfect result for a road with several kilometers, but it may also mean a sudden dangerous barrier, a bump or merely an elevation measurement bias.

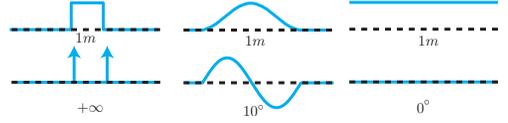


Fig. 4. Examples on Accuracy for Road Shape

In order to overcome the shortcomings, we propose the road slope as the second metric for the road modeling accuracy. Slope grade is an essential factor for road and is suggested by several well studied literatures [8] [9]. Thus the maximum angle error should be smaller than the required threshold:

$$\max_{i \in \{1 \dots n\}} |\theta_m(s_i) - \theta(s_i)| \leq \beta \quad (5)$$

In the lower part of Fig.4, three situations can be distinguished by the maximum angle error.

## 3.2 Frame Transformation Calculation I

This section introduces our first algorithm that uses accelerometer and magnetometer to determine the reading transformation matrix from b-frame to n-frame. In another word, we want to determine the mobile phone position in 3D space with respect to the Earth. This calculation is straightforward once we take the gravity acceleration direction and the magnetic North direction into consideration. Both of their readings are constant in the Earth frame but are flexible in the body frame. The position is calculated by quantifying the difference between the two groups of readings. However, the simple calculation does not guarantee accuracy under different measurement conditions. Our contribution is to make use of different sensor properties to minimize the total calculation error. We first describe the basic calculation method, then we propose the compensation methods.

### 3.2.1 Calculation Method

A same vector has different readings in different frames. For example, the moving direction in Fig.1 is a constant vector in b-frame but a variable in n-frame. The transformation matrix  $C_b^n$  transforming the vector reading  $\vec{v}^b$  in b-frame to the n-frame reading  $\vec{v}^n$  is a  $3 \times 3$  orthonormal matrix. Its orthogonality  $C_b^n = (C_b^n)^{-1} = (C_b^n)^T$  makes the nine unknowns less flexible. Actually, one needs at least four readings  $\vec{v}_1^n, \vec{v}_1^b, \vec{v}_2^n, \vec{v}_2^b$  from two different vectors  $\vec{v}_1, \vec{v}_2$  to uniquely determine  $C_b^n$ . Those two groups of readings can be measured by accelerometer and magnetometer respectively.

The first group of readings is provided by accelerometer, whose reading  $\vec{A}(t)$  is a measurement of the supporting force  $\vec{F}_{supp}^b(t)$  that the mobile phone applies on its accelerometer chip with respected to b-frame.  $C_n^b(t)$  is the transformation matrix that we want to solve.  $\vec{F}_{supp}^n(t)$  is the supporting force reading with respect to n-frame.  $\delta_{\vec{A}}(t)$  is the measurement error. In addition to  $\vec{F}_{supp}^n(t)$ , the gravity force  $\vec{G}$  also applies on the chip. Consider the synthesis force  $\vec{F}^n(t) = \vec{F}_{supp}^n(t) + \vec{G}^n$ , we can get the acceleration of the mobile phone in n-frame  $\vec{a}_{nb}^n(t) = \vec{F}^n(t)/m_0$ . The synthesis force value equals to the acceleration with unit mass, then the accelerometer reading can be expressed as:

$$\begin{aligned}\vec{A}(t) + \delta_{\vec{A}}(t) &= C_n^b(t) \cdot \vec{F}_{supp}^n(t) = C_n^b(t) \cdot (\vec{F}^n(t) - \vec{G}^n) \\ &= C_n^b(t) \cdot (\vec{a}_{nb}^n(t) - \vec{g}^n)\end{aligned}\quad (6)$$

Particularly, when the phone is at rest or in uniform motion,  $\vec{a}_{nb}^n(t) = 0$ , we have this estimate:

$$\vec{A}(t) \approx C_n^b(t) \cdot -\vec{g}^n \quad (7)$$

The second group of readings is provided by magnetometer, whose reading  $\vec{M}(t)$  is the measurement of the strength and direction of magnetic fields with respect to b-frame:

$$\vec{M}(t) + \delta_{\vec{M}}(t) = C_n^b(t) \cdot \vec{M}_e^n + \vec{M}_l^b(t) \quad (8)$$

$\delta_{\vec{M}}(t)$  is the measurement error. The Earth's magnetic field  $\vec{M}_e^n$  is ubiquitous and approximately points to the North. Its exact value is provided in the public database [10].  $\vec{M}_l^b(t)$  is generated by the local surrounding materials, such as Iron product, electric current and so on. Particularly, when there is no local magnetic interference,  $\vec{M}_l^b(t) = 0$ , we have the estimate:

$$\vec{M}(t) \approx C_n^b(t) \cdot \vec{M}_e^n \quad (9)$$

From the readings of the two sensors,  $C_n^b(t)$  can be estimated by solving several equations from (7)(9).

### 3.2.2 Accelerometer Compensation

The real measurement is far from the ideal assumptions in (7). For the accelerometer, there are two reasons affecting the calculation of  $C_n^b(t)$ . First, the car speed is always changing, so the acceleration of the mobile phone body with respect to n-frame is not zero:  $\vec{a}_{nb}^n(t) \neq 0$ . We compensate it by considering the horizontal acceleration obtained from GPS. Second, the value of  $\delta_{\vec{A}}(t)$  is too large to be ignored. We unify the accelerometer reading to eliminate the impact of scalar value error in solving  $C_n^b(t)$ . Details are presented below.

First, from (6),  $\vec{a}_{nb}^n(t)$  is unknown and can not be determined by accelerometer. We solve this problem by introducing the GPS horizontal acceleration, which is calculated through the second

derivative of the horizontal locations:  $\vec{a}_{GPS||}^n(t) = [d^2c_x(t)/dt^2, d^2c_y(t)/dt^2, 0]^T$ . If we divide  $\vec{a}_{nb}^n(t)$  into horizontal part and vertical part  $\vec{a}_{nb}^n(t) = \vec{a}_{nb||}^n(t) + \vec{a}_{nb\perp}^n(t)$ ,  $\vec{a}_{GPS||}^n(t)$  is an estimate for  $\vec{a}_{nb||}^n(t)$ . Moreover, the time interval between two sample points is small (0.02s in our implementation, sensors by resampling, GPS at 1Hz by interpolation §4), so we can use the transformation matrix  $C_n^b(t^-)$  in the previous sample to approximate  $C_n^b(t)$ . The new acceleration relation is:

$$\begin{aligned}\vec{A}(t) &= \vec{A}(t) - C_n^b(t^-) \cdot \vec{a}_{GPS||}^n(t) \\ &= C_n^b(t) \cdot (\vec{a}_{nb}^n(t) - \vec{g}^n) - C_n^b(t^-) \cdot \vec{a}_{GPS||}^n(t) - \delta_{\vec{A}}(t) \\ &\approx C_n^b(t) \cdot (\vec{a}_{nb\perp}^n(t) - \vec{g}^n)\end{aligned}\quad (10)$$

Second, by doing tests, we find  $\delta_{\vec{A}}(t)$  affects the amplitude of  $\vec{A}(t)$ . The measured  $|\vec{A}(t)|$  differs from 9.4 to 10.4 in different positions. If  $|\vec{A}(t)| \neq |\vec{g}^n| \approx 9.8$ , the derived  $C_n^b(t)$  will not satisfy the orthogonality. On the other hand, there still has an unknown  $\vec{a}_{nb\perp}^n$  in (10), whose direction is same as  $\vec{g}^n$ . We keep the orthogonality and eliminate  $\vec{a}_{nb\perp}^n$  through normalizing both sides of the equation in (10).

$$\vec{A}(t)/|\vec{A}(t)| \approx C_n^b(t) \cdot [0, 0, 1]^T \quad (11)$$

### 3.2.3 Magnetometer Compensation

The local magnetic field  $\vec{M}_l^b(t)$  is a main contributor to the magnetometer readings in many cases, where (9) does not hold. We treat  $\vec{M}_l^b(t) = \vec{M}_l^b$  as an unknown constant parameter and calculate it during the learning step. Readings of the magnetometer are specially rotated and scaled to reduce error. We present these two approaches below.

First, any electric current and magnetic material near the mobile phone generate magnetic fields, which contribute to  $\vec{M}_l^b(t)$ . Fig.5(a) is an in-door test. The mobile phone is hanged over a steel table from 70cm to 0cm.  $\vec{M}(t)$  is recorded with different distance values and shown in dot points. The color of the point indicates the distance; the darker the closer. The black arrow is the negative gravity direction, obtained by averaging accelerometer readings. The angles between the black arrow and magnetometer readings  $\langle \vec{A}(t), \vec{M}(t) \rangle \approx \langle -\vec{g}^n, \vec{M}_e^n \rangle$  are constant if  $\vec{M}_l^b(t) = 0$ . The result shows that  $\vec{M}_l^b(t)$  is definitely none zero and is a main component of the magnetic fields in common cases.

The experiment in Fig.5(b) illustrates the possibility to compensate  $M_l^n(t)$ . We put the mobile phone screen up on a chair that can rotate, and the sensor readings are recorded at different angles until the rotation forms a circle. The black arrow is the negative gravity direction obtained by averaging  $\vec{A}(t)$ . The hollow points are the readings of the magnetometer and form a circle. However, due to the impact of the material of the chair and the mobile phone case, the center of

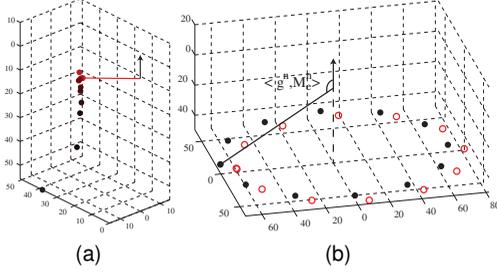


Fig. 5. Magnetometer Test

the circle is slightly shifted from the expected value and violates the angle relation  $\langle -\vec{g}^n, \vec{M}_e^n \rangle$ . We explore the relation between the two circles and find the local magnetic field  $M_l^b = C_n^b(t) \cdot M_l^n(t)$  is a constant<sup>3</sup> and can be estimated through a closed loop path. The compensation value  $\vec{M}_{estl}^b$  is generated as user profile in §3.5.1.

$$\vec{M}'(t) = \vec{M}(t) + \vec{M}_{estl}^b \approx C_n^b(t) \cdot \vec{M}_e^n \quad (12)$$

Second, the angle constraint does not hold for every sample in (11) and (12) due to various errors:  $\langle \vec{A}'(t), \vec{M}'(t) \rangle \neq \langle -\vec{g}^n, \vec{M}_e^n \rangle$ . Those error could originate from in-car magnetic field changes caused by electronic devices or out-car magnetic field changes caused by different terrains and buildings [11]. This inequality makes  $C_n^b(t)$  violate the orthogonality. The problem is relieved by adjusting the direction of  $\vec{A}'(t)$  or  $\vec{M}'(t)$ . We choose to adjust  $\vec{M}'(t)$  because  $\vec{M}_{estl}^b$  contains more error sources for two reasons:  $\vec{M}_l^b(t)$  is not a strict constant due to several unpredictable factors such as the mobile phone current, and the rotation in closed loop does not form a perfect circle because of the road slope angle variation.

We rotate  $\vec{M}'(t)$  in the plane determined by  $\vec{A}'(t)$  and  $\vec{M}'(t)$ . The new relation is obtained by vector cross product:

$$\vec{M}_\perp = (\vec{A}'(t) \times \vec{M}'(t)) \times \vec{A}'(t) \quad (13)$$

$$\vec{M}_\perp / |\vec{M}_\perp| \approx C_n^b(t) \cdot [0, 1, 0]^T \quad (14)$$

### 3.2.4 Method Summary

The calculation of the frame transformation matrix  $C_n^b(t)$  uses the sensing data from accelerometer, magnetometer and GPS. The method flow is:

- 1) The original accelerometer reading  $\vec{A}(t)$  is compensated in (10) with the horizontal acceleration estimated from GPS trace.
- 2) The original magnetometer reading  $\vec{M}(t)$  is compensated in (12) with the local magnetic field bias  $\vec{M}_{estl}^b$ , which is a constant after the learning step.

3. As the strength of the magnetic field attenuates inversely with the third power of the distance, which is considered to be very fast. The main contributor for  $\vec{M}_l^b(t)$  is the closely attached vehicle body. Other magnetic interference from the nearby mobile devices such as notebook is negligible after tens of cm attenuation.

- 3) We rotate  $\vec{M}'(t)$  according to  $\vec{A}'(t)$  in (13) to meet the angle constraint that restricts the direction of the gravity and geomagnetic fields.
- 4) We normalize obtained vectors to reduce scalar value error in (11) and (14)
- 5) The third column of  $C_n^b(t)$  is estimated by (11). The second column of  $C_n^b(t)$  is estimated by (14). The first column is determined by the other two columns through the orthogonality.

The composition of  $C_n^b(t)$  shows the essence of the calculation. Consider the moving direction vector reading in b-frame  $\vec{d}^b$ , the corresponding reading  $\vec{d}^n(t) = C_n^b(t) \cdot \vec{d}^b$  in n-frame is the slope angle that we want to obtain. As  $C_n^b(t) = (C_n^b)^T(t)$ ,  $d_z^n(t)$  is determined by the third column of  $C_n^b(t)$ , which is mainly measured by the accelerometer. The similar observation holds for  $d_y^n(t)$ , which is mainly determined by the magnetometer. The conclusion is that the accuracy of the slope angle calculated in this method is mainly determined by the accelerometer.

## 3.3 Frame Transformation Calculation II

In this section, we use sensing data from gyroscope to calculate  $C_n^b(t)$ . §3.2 calculates  $C_n^b(t)$  at very  $t$  independently, while the calculation in this section based on the change rate of  $C_n^b(t)$ . We first introduce the basic algorithm, then we discuss the calibration method.

### 3.3.1 Calculation Method

Gyroscope is a sensor that measures the mobile phone body rotation angular velocity. Its reading  $\vec{W}(t)$  estimates the rotation angular velocity  $\vec{\omega}_{nb}^b(t)$  of the mobile phone b-frame with respect to n-frame. In Fig.2, the circular arrow is the positive rotation direction. For example, if we put the phone screen up on the desk and rotate it counterclockwise, the z-axis reading  $W_z$  will have positive value.

The relation [12] between  $C_n^b(t)$  and  $\vec{\omega}_{nb}^b(t)$  is:

$$\begin{aligned} \frac{dC_n^b(t)}{dt} &= C_n^b(t) \cdot \begin{bmatrix} 0 & -\omega_{nbz}^b(t) & \omega_{nby}^b(t) \\ \omega_{nbz}^b(t) & 0 & -\omega_{nbx}^b(t) \\ -\omega_{nby}^b(t) & -\omega_{nbx}^b(t) & 0 \end{bmatrix} \\ &= C_n^b(t) \cdot T(\vec{\omega}_{nb}^b(t)) \end{aligned}$$

In discrete case, take the previous sampling time as  $t^-$ :

$$C_n^b(t) = C_n^b(t^-) \cdot \left( T(\vec{\omega}_{nb}^b(t))(t - t^-) + I \right) \quad (15)$$

In (15),  $C_n^b(t)$  is calculated in a iterative way. The initial value  $C_n^b(t_0)$  is obtained through the method in §3.2.

### 3.3.2 Gyroscope Calibration

Consider the simple error model [12] for x component in gyroscope  $\alpha_x \omega_{nbx}^b + \beta_x = W_x$ , where  $\beta_x$  is the bias

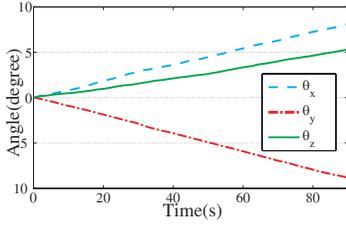


Fig. 6. Static Gyroscope Measurement

error and  $\alpha_x$  is the scale error. We perform one experiment to show the error properties of gyroscope, and propose a simple scheme to reduce the bias error.

In the experiment, we record the reading of the gyroscope in a static situation<sup>4</sup>. Results are shown in Fig.6. The angle value is the integration of the corresponding angular velocity, for example,  $\theta_x(t) = \int_0^t \omega_x(t_i) dt_i$ . This results shows that the low-cost gyroscope has large bias error  $\beta$ . The value of  $\beta$  is stable because the slope of the curve does not change much. Therefore, we treat  $\beta$  as a constant and update it whenever the mobile phone is not moving. The static period can be judged by the variance of the acceleration. The gyroscope is calibrated in (16).

$$\vec{W}'(t) = \vec{W}(t) - \vec{\beta} \approx \vec{\omega}_{nb}^b(t) \quad (16)$$

### 3.3.3 Method Summary

The calculation method based on gyroscope works as:

- 1) The gyroscope bias error  $\vec{\beta}$  is updated according to (16) once the mobile phone is static.
- 2) We calculate one initial matrix  $C_b^n(t_0)$  from §3.2.4.
- 3)  $C_b^n(t)$  is calculated iteratively by taking (16) into (15).

From (15), the measurement error in gyroscope accumulates with time. We will handle this problem in the next section.

## 3.4 3D Road Modeling

Guided by the accuracy metrics, the 3D road model of *3Rd* contains two aspects: the road height and the road slope angle. We first obtain the road slope angle  $\theta(s)$  by using the the moving direction and the frame transformation matrix obtained in §3.2 and §3.3. Then we generate the road height using the geometric relation between the road angle and horizontal displacement.

### 3.4.1 Road Slope Angle Calculation

Given the frame transformation matrix  $C_b^n(t)$  and the moving direction  $\vec{d}^b$  with respect to the car, the moving direction with respect to the Earth is given by

4. n-frame is not an inertial frame actually. The angular rotation speed of Earth in inertial space is about  $4.2 \times 10^{-3}$  dps(degree/second). The impact is very small in our experiments

$\vec{d}^n(t) = C_b^n(t) \cdot \vec{d}^b$ . Then,  $\vec{d}^n(t)$  is transformed to  $\vec{d}^n(s)$  by aligning the time in GPS trace as (1). Road slope angle  $\theta(s)$  is determined by  $\vec{d}^n(s)$  in (3). We contribute to this simple calculation by fusing the measurement data from different sensor sources.

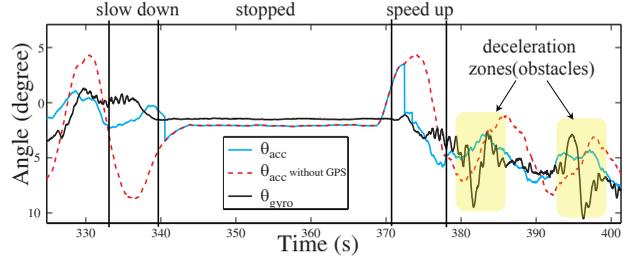


Fig. 7. Angle Measurement Comparison

The fusing method is motivated by the properties of the two different sensor sources used to derive  $C_b^n(t)$ : The one in §3.2 is mainly based on accelerometer and the one in §3.3 is mainly based on gyroscope. We use  $\theta_{acc}(t)$  and  $\theta_{gyro}(t)$  to denote the calculated angle respectively. Consider the motivated example in Fig.7 taken from the real trace in §5, it shows the real performance of  $\theta_{acc}(t)$  and  $\theta_{gyro}(t)$ . The car is first stopped by the traffic light. After several seconds, the car speeds up and encounters two deceleration zones (Fig.10(b)). Those curves indicate three points: First,  $\theta_{acc}(t)$  and  $\theta_{gyro}(t)$  are quite different wherever acceleration occurs. Second, the difference is not caused by the GPS compensation scheme in §3.2.2. On the contrary, one reason for the difference is that our compensation scheme can not fully compensate the horizontal acceleration. Third,  $\theta_{gyro}(t)$  shows a fine variation when the car encounters the deceleration zones. This example motivates us to take  $\theta_{gyro}(t)$  as the complement to describe the road angle detailed variations.

In order to leverage the properties in the modeling, we verify them in distance domain. Consider  $\theta_{acc}(s) = \theta(s) + \delta_{acc}(s)$  and  $\theta_{gyro}(s) = \theta(s) + \delta_{gyro}(s)$ , where  $\theta(s)$  is the real angle and  $\delta_{acc}(s)\delta_{gyro}(s)$  are the measurement errors. First, if we treat  $\delta_{acc}(s)$  as a random variable dominated by car acceleration, its variation is of high frequency compared with the road shape such as down hill or up hill. In other words, the low-frequency part of  $\theta_{acc}(s)$  is accurate. Second,  $\theta_{gyro}(s)$  is not affected by the error sources in  $\theta_{acc}(s)$  and the random noise in gyroscope are eliminated by the integration-like calculation. However,  $\delta_{gyro}(s)$  is caused by the system error of gyroscope and increases with time/distance. Therefore,  $\delta_{gyro}(s)$  is of low frequency and the high frequency part of  $\theta_{gyro}(s)$  is accurate.

From the above analysis, we design two filters to improve the calculation for  $\theta(s)$ . We use a low-pass filter on  $\theta_{acc}(s)$  to remove  $\delta_{acc}(s)$  and the high-frequency part of  $\theta(s)$ . Similarly, we use a high-

pass filter on  $\theta_{gyro}(s)$  to remove  $\delta_{gyro}(s)$  and the low-frequency part of  $\theta(s)$ . The filtered signals are combined to get the new estimate  $\theta_m(s)$ . The cut-off frequency we selected is  $200m$ , which means an uphill and downhill pair is treated of high frequency if their horizontal distance is less than  $200m$ .

### 3.4.2 Road Curve Calculation

We generate the road curve model  $c_{mz}(s)$  by considering the geometric relation in (2) and (3). In discrete case, it is:

$$c_{mz}(s_k) = \sum_{i=0}^k (s_{i+1} - s_i) \tan \theta_m(s_i) \quad (17)$$

### 3.4.3 Method Summary

- 1) We use  $C_n^b(t)$  from §3.2 §3.3 and  $\vec{d}^b$  from user profile to get two road slope angle estimates  $\theta_{acc}(t)$  and  $\theta_{gyro}(t)$ .  $\theta_{acc}(t)$  and  $\theta_{gyro}(t)$  are mapped to  $\theta_{acc}(s)$  and  $\theta_{gyro}(s)$  according to the GPS trace(1).
- 2) We fuse the two slope models by applying band pass filters:  $\theta_m(s) = Lowpass(\theta_{acc}(s)) + Highpass(\theta_{gyro}(s))$ .
- 3) The vertical dimension of the road curve  $\vec{c}(s)$  is given by  $c_{mz}(s)$  in (17)

## 3.5 Closed Loop Learning

Mobile phone, as a personal device, incurs measurement diversity. We design a learning step in  $3Rd$  to unify the diversity while keep their manual adjustment minor. The diverse factors, specifically, the position of the mobile phone holder and the in-car magnetic field, are relatively stable, so we treat them as the user profile and obtain them before the first use. Since the user only need to perform this step once per car with a fixed mobile phone holder, we believe this learning process is of low overhead.

The learning process is finished by driving along a path satisfying two requirements: First, the driving starts at a certain direction and ends at a similar direction. This requirement makes the vehicle rotate for a circle and the in-car magnetic field can be calculated §3.2.3. Second, the path should be self-crossed. The crossing points have the same height and can be used as a constraint to calculate moving direction  $\vec{d}^b$ . A closed loop path concurrently satisfies the two requirements, so that it is chosen in the learning step of  $3Rd$  design.

### 3.5.1 Learning: Local Magnetic Fields Calculation

We calculate  $\vec{M}_l^b(t)$  based on two facts. One is that the strength of the magnetic field inversely proportional to the cubic of the distance to the source, so  $\vec{M}_l^b(t)$  is dominated by the source nearby. As the mobile phone is fixed on the car, the magnetic sources such as car body change directions the same as the mobile phone. In the view point of the mobile phone, the

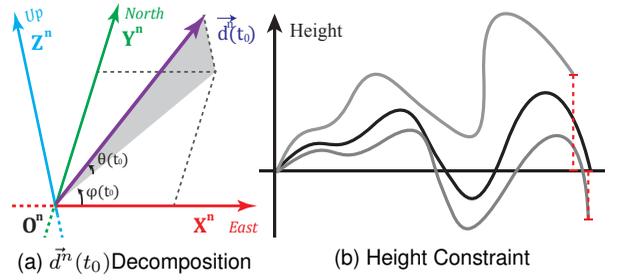


Fig. 8. Moving Direction  $\vec{d}^b$  Calculation

local magnetic field with respect to the phone body is constant during the measurement:  $\vec{M}_l^b(t) = \vec{M}_l^b$ . Another fact is that the car body rotates for a circle in the close loop due to the alignment of the start and end points. Similar to Fig.5(b), this fact indicates  $\vec{M}_e^b(t) = C_n^b(t) \cdot \vec{M}_e^n$  forms a circle.

Consider the magnetometer readings  $\vec{M}(t) = \vec{M}_e^b(t) + \vec{M}_l^b - \delta_{\mathcal{M}}(t)$ , it is a circle with a constant center bias  $\vec{M}_l^b$  from circle  $\vec{M}_e^b(t)$ . Note that the center and radius of  $\vec{M}_e^b(t)$  satisfy the angle constraint  $\langle \vec{A}(t), \vec{M}_e^b(t) \rangle \approx \langle -C_n^b(t) \cdot \vec{g}^n, C_n^b(t) \cdot \vec{M}_e^n \rangle = \langle -\vec{g}^n, \vec{M}_e^n \rangle$ . Hence, we use a least squares method [13] to move  $\vec{M}(t)$  to meet the angle constraint. The moved value  $\vec{M}_{estl}^b$  is the estimate of  $-\vec{M}_l^b$ .

### 3.5.2 Learning: Moving Direction Calculation

The key idea for calculating the local moving direction  $\vec{d}^b$  is shown in Fig.8. The initial value of  $\vec{d}^b(t_0) = C_b^n(t_0) \cdot \vec{d}^b$  is divided into two parts. The horizontal part  $\varphi(t_0)$  determines the horizontal model shape and the vertical part  $\theta(t_0)$  determines the vertical variation. We search  $\varphi(t_0)$  by minimizing the error between the horizontal model and the GPS trace. After that, the curve shape is almost determined, then we fix  $\varphi(t_0)$  and search  $\theta(t_0)$  to make the height difference between the self-cross points as small as possible.

Specifically, given the sampling time  $t_0 \dots t_n$  and the self-crossed points at  $t_0$  and  $t_n$ , The scheme works as:

- 1) Searching is started by decomposing  $\vec{d}^b$  into a function of  $\varphi(t_0)$  and  $\theta(t_0)$ :

$$\begin{aligned} \vec{d}^n(t_k) &= C_b^n(t_k) \cdot \vec{d}^b = C_b^n(t_k) C_n^b(t_0) \cdot \vec{d}^n(t_0) \\ &= C_b^n(t_k) C_n^b(t_0) \cdot \vec{v}^n(\theta(t_0), \varphi(t_0)) \end{aligned} \quad (18)$$

The initial rough estimates  $\varphi_m(t_0)\theta_m(t_0)$  for  $\varphi(t_0)\theta(t_0)$  are easily obtained by considering the car acceleration.

- 2) Takes the horizontal displacement between  $t_i$  and  $t_{i+1}$  as  $\Delta s(t_i) = s(t_{i+1}) - s(t_i)$ . The horizontal model is given by (19).  $\varphi_m(t_0)$  is searched to minimize the total horizontal difference between the horizontal model and the GPS trace:

$$c_{mx|y}(t_k) = \sum_{i=0}^k \Delta s(t_i) d_{x|y}^n(t_i) / \sqrt{d_x^{n^2}(t_i) + d_y^{n^2}(t_i)} \quad (19)$$

$$\varphi_m(t_0) = \arg \min_{\varphi(t_0)} \sum_{i=0}^n \sqrt{(c_{mx}(t_i) - c_x(t_i))^2 + (c_{my}(t_i) - c_y(t_i))^2} \quad (20)$$

3) We fix  $\varphi(t_0)$  to  $\varphi_m(t_0)$  and search  $\theta(t_0)$  to minimize the height difference at the crossing point:

$$\theta_m(t_0) = \arg \min_{\theta(t_0)} |c_{mz}(t_n) - c_{mz}(t_0)| \quad (21)$$

The moving direction with respect to the mobile phone body  $\vec{d}^b$  is obtained by substituting  $\varphi_m(t_0)$  and  $\theta_m(t_0)$  into (18). Due to the space limitation, we omit the proof of the effectiveness and the uniqueness of the searching method for  $\varphi(t_0)$  and  $\theta(t_0)$ .

## 4 IMPLEMENTATION

The data is collected by Samsung Galaxy S II with Android 2.3. We use the FASTEST level of sampling rate in the sensing data logging application. With all the sensors opening, the sampling rate of the accelerometer and the magnetometer is about 60Hz and the rate of the gyroscope is about 100Hz. Each record is triggered by one sensor event and has the format  $\langle timestamp, v_x, v_y, v_z \rangle$ . *timestamp* is the event time in nanosecond and its value is counted from the last power on. GPS has an independent mechanism, whose fastest sampling rate is fixed to 1 Hz. The default timing method in GPS uses milliseconds counted from 1970. We record a nanosecond time once GPS event triggers the handle in our program to align the time with sensor records. The record for GPS has similar format  $\langle timestamp, latitude, longitude, altitude \rangle$ .

3Rd server algorithm is currently implemented in Matlab without parallelism. The desktop is Intel i7 950 and 3G RAM. The total modeling time for a 2km road is about 6.1s, which can be largely improved with C implementation.

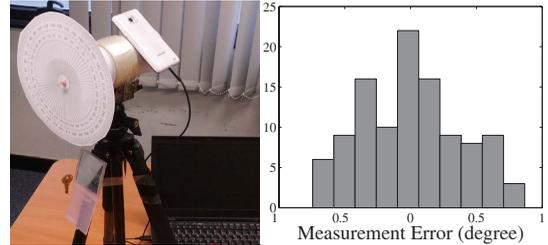
In order to unify the sampling time among sensors, we use 50Hz to resample the sensing data. Normally, the car speed is less than 25m/s or 90km/h. 50Hz is enough to sense the slope changes within 1m. We use the linear interpolation in the resampling, alternative interpolation methods do not show improvement in performance. After resampling, we obtain  $\vec{M}(t)$ ,  $\vec{A}(t)$  and  $\vec{W}(t)$ . For GPS, its sampling rate is much lower than 50Hz. The horizontal trace is linearly interpolated and smoothed by a moving-sum-filter, and then we have  $c_x(t), c_y(t)$ . The above all are the complete input for the modeling method in §3.

## 5 EVALUATION

We evaluate 3Rd in four steps. First we give a field solution to accurately measure the slope in real road as the benchmark. Then, we compare the results of 3Rd to the benchmark data. After that, we explain the limitation of existing 3D road modeling methods. Last, a real application example is used to show the effectiveness of 3Rd.



(a) Slope Measurement Device



(b) Accuracy Evaluation

(c) Results

Fig. 9. Slope Measurement Device

### 5.1 Real Road Benchmark

In this subsection we describe the benchmark used for evaluation. First we introduce the field measurement method, then we describe the properties of the obtained benchmark.

#### 5.1.1 Road Slope Measurement

In order to get the benchmark data from real roads, a field measurement device is developed based on the accelerometer. The mechanism of this device is using the accelerometer to measure the difference of the gravity directions between the sampling location and the level point.

Fig. 9(a) shows its architecture. The mobile phone is tightly fixed on a cubic box by scotch tapes. The box has large underside to tolerant sands and small stones on the road surface. There is another white cube between the box and the mobile phone. The gap between the white cube and the box is filled by papers to adjust fine-grained level. We use a ping-pong ball to find a level desk and record the level point reading. In the measurement, we put down the device on the center of the road and head it to the moving direction. The angle between the level point and the acceleration direction is treated as the slope angle  $\theta$ . Then, we use GPS and a tape to map the angles to their locations. GPS Locations are measured every 10m. Road angles are measured every meter and equally inserted between two GPS locations. Finally, we get  $\theta(s)$ . Height  $c_z(s)$  is obtained by (17).

As the accuracy of the benchmark is determined by the accelerometer. We experimentally quantify the direction measurement ability of the accelerometer in Fig. 9(b)(c). The evaluation is done through comparing the measured gravity direction by the accelerometer with the result obtained by a protractor. As shown in Fig. 9(b), the mobile phone is fixed to a protractor panel by a solid cylinder. The gravity direction is indicated by a pendant. A mirror is used to aid accurate

reading. When the mobile phone rotates for  $\psi$ , which can be read from the pendant, the gravity direction measured by the accelerometer should change for a same angle.

Based on  $\psi$ , the mobile phone is rotated for  $360^\circ$  in  $10^\circ$  interval. The rotated angle measured by the accelerometer is defined as the difference between the initial gravity direction and current gravity direction:  $\psi'(i) = \langle \vec{A}(0), \vec{A}(i) \rangle$ . Therefore,  $\psi'(i+1) - \psi'(i) - 10^\circ$  is just the error of angle measurement. The same test is done for three axes, the overall histogram is shown in Fig. 9(c). Results indicate the error of the slope angle by this device is within  $\pm 0.8^\circ$ , which is the error bound for our benchmark.

### 5.1.2 The Benchmark

The road for evaluation is shown in Fig. 10(a). The benchmark is measured on the road with the tool described in Fig. 9(a). Results are shown in Fig. 11(b)(c) tagged with “benchmark”.

The reason for choosing this road is the rich slope variation and road conditions. In a high view, the road is a 1.7km mountainous loop road. It consists of one downhill (300 – 900m), one uphill (900 – 1700m) and several plane segments. The average slope angle of the downhill and the uphill is about  $6^\circ$ , which is definitely considered as steep slope [14] and several warning signed are set along the road. In addition to the steep slope trend, there exists 8 deceleration zones whose surface are raised to keep drivers alert. Their variation contribute to the 8 peaks in Fig.10(b). Further, the road also contains several bumps in the segment under construction(50 – 300m). As a summary, the CDF in Fig.10(b) shows that the slope variation in 10m is larger than  $3^\circ$  in more than 10% of the road segment. The above features enrich road conditions and make it suitable for modeling evaluation.

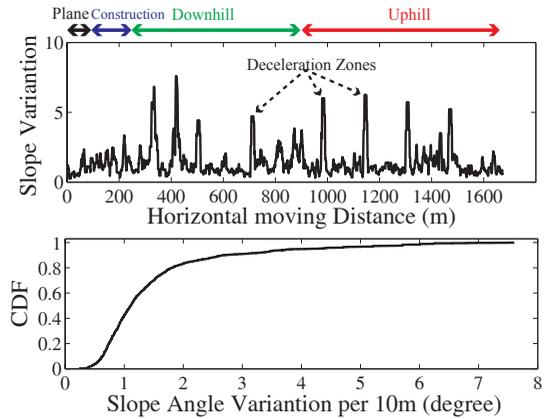
The accuracy of the slope angle in the benchmark is quantified in §5.1.1. For the height benchmark, we view its accuracy indirectly. First, the last point of the loop road is just the beginning of the road. If the initial point is 0, then the last point  $c_z(s_n)$  is calculated by accumulated summation and equals to 0 in principle. The actual value of  $c_z(s_n)$  is 0.7m. Further, the difference of the maximum and minimum altitude of the road consulted from road construction map is  $137.64 - 72.85 = 64.79$ m. The same value in our measurement is 63.97m, and their difference is 0.82m. The above two numbers indicate the height error in the benchmark is within one meter.

## 5.2 Real Road Evaluation

In this subsection, we compare the modeling result of *3Rd* with the benchmark and evaluate component schemes separately.



(a) The Road for Evaluation



(b) Road Conditions



(c) Deceleration Zone in Yellow

Fig. 10. The Benchmark Road

### 5.2.1 3D Modeling Overview

We perform *3Rd* method on the road in Fig.10(a). The mobile phone is fixed on the taxi by tape or mobile phone holder. We ask the driver to drive along the road for several loops. In the first loop, we perform the learning step to generate the user profile (the driving direction and the magnetic noise). We use the data in other loops to model the road and show the effectiveness of the method. The modeling tests are performed for 17 times with different taxis and mobile phone positions. The driving speed for each test is very diverse depending on the traffic light and pedestrians. The statics of the speed for all the tests are shown in Fig.12. The portion of zero speed is mainly caused by the traffic light in the construction segment. During 70% of the time, the speed is less than 40km/s, which is the normal speed in urban areas. The maximum speed is 60km/s, which is close to the designed speed of highways of the mountainous terrain [15].

The final model of the road is shown in Fig.11. Based on the metrics in §3.1.3, we view the model in

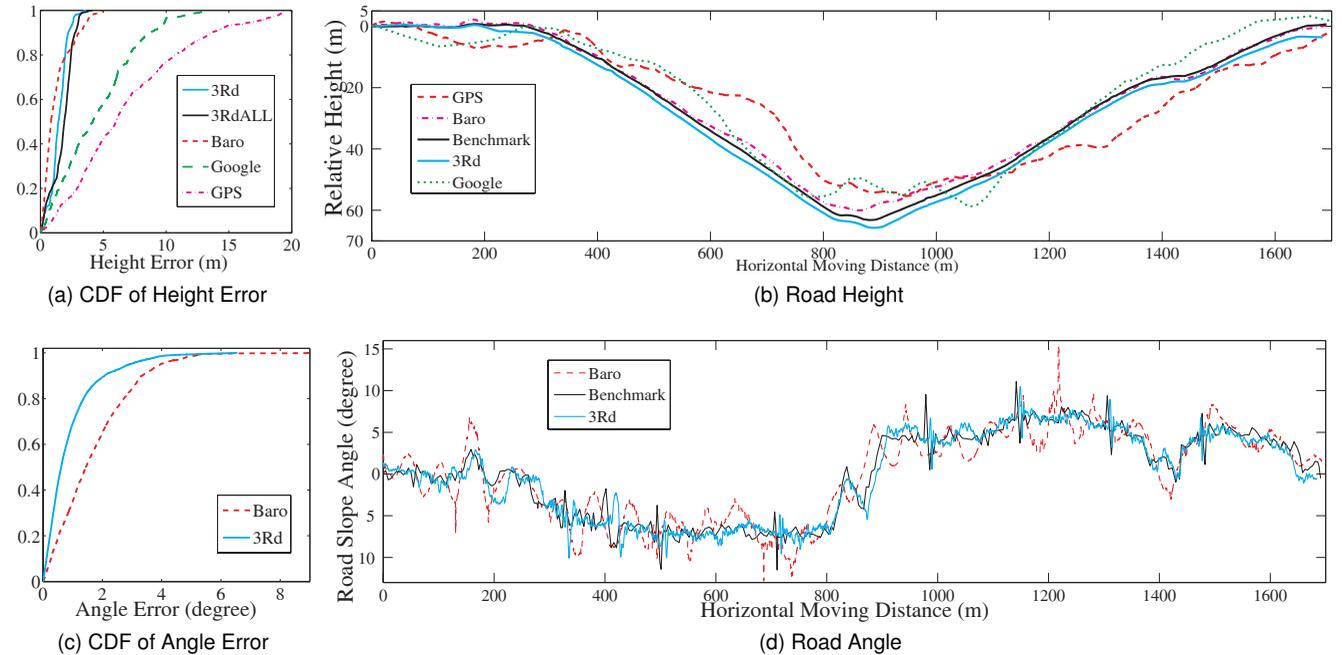


Fig. 11. Real Road Evaluation

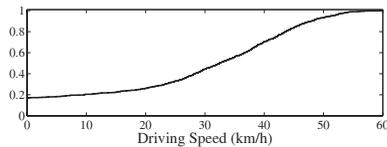


Fig. 12. Driving Speed of the Modeling Evaluation

both height and angle. CDF of the height error for all the tests is tagged as *3RdALL*. CDF of the angle error for all the tests is coincidence with the one with tag *3Rd*. As the results for different tests are quite similar, curves in Fig.11 tagged as *3Rd* only present the results of one test. The modeling results are compared with the benchmark in Fig. 11. The maximum height error is about  $4m$ . The mean height error is  $1.65m$ . 90% height errors are less than  $2.5m$ . The maximum angle error is about  $6.5^\circ$ , 90% angle errors are less than  $2^\circ$ .

### 5.2.2 Components Evaluation

In *3Rd*, we develop and combine several schemes to improve the modeling accuracy. We evaluate each component by disabling it to see the modeling result. Table.1 lists the brief summery and the corresponding abbreviations in Fig.13. we use *3Rd-X* to indicate "3Rd minus X scheme". For example, *3Rd-2* denotes the *3Rd* modeling results without compensating the magnetic field interference. We ignore *3Rd-4* in this section because of its obvious results. All the component evaluation share the same user profile unless otherwise mentioned.

In Fig. 13(a), we zoom in the last several meters of Fig. 11(d) to show the impact of horizontal acceleration without compensation. The upper part

TABLE 1  
Method Components

	Section	Method	Abbr.
1	§3.2.2	Reduce the horizontal acceleration $\vec{a}_{GPS  }^n(t)$ through GPS	<i>3Rd-1</i>
2	§3.2.3	Compensate the magnetometer reading with local magnetic field $\vec{M}_l^b$	<i>3Rd-2</i>
3	§3.2.3	Eliminate the impact of magnetometer on angle measurement by rotating	<i>3Rd-3</i>
4	§3.3.2	Subtract the bias error from gyroscope	<i>3Rd-4</i>
5	§3.4.1	Fuse the measurement of two sensors through filter	<i>3Rd-5</i>

of Fig.13(a) is the horizontal acceleration calculated from GPS. The car is stopped at  $1700m$ , so the latter part of the curve shows a continuous deceleration. Measured angle by *3Rd-1* in the lower part of Fig.13(a) is distorted from the benchmark with a trend similar as the horizontal acceleration.

As we directly fix the mobile phone on the platform under the windshield of the taxi, the local magnetic field  $\vec{M}_l^b$  is large. In Fig. 13(b), points tagged as *3Rd-2* are the magnetometer readings without compensation. The black arrow is the average of the negative gravity direction. The magnetic compensation scheme plays critical role in generating accurate moving direction. The best fitted curve in horizontal searching in §3.5.2 is shown in Fig. 13(c). Due to the wrong horizontal moving direction, the vertical moving direction searching is also affected. Take the wrongly selected moving direction  $\vec{d}^b$ , the real measurement shows large error in Fig.13(e)(f) tagged *3Rd-2*.

In Fig. 13(e)(f), lines tagged *3Rd-3* are generated by rotating the  $\vec{A}(t)$  to fit the angle restriction  $\langle -\vec{g}^n, \vec{M}_e^n \rangle$ .

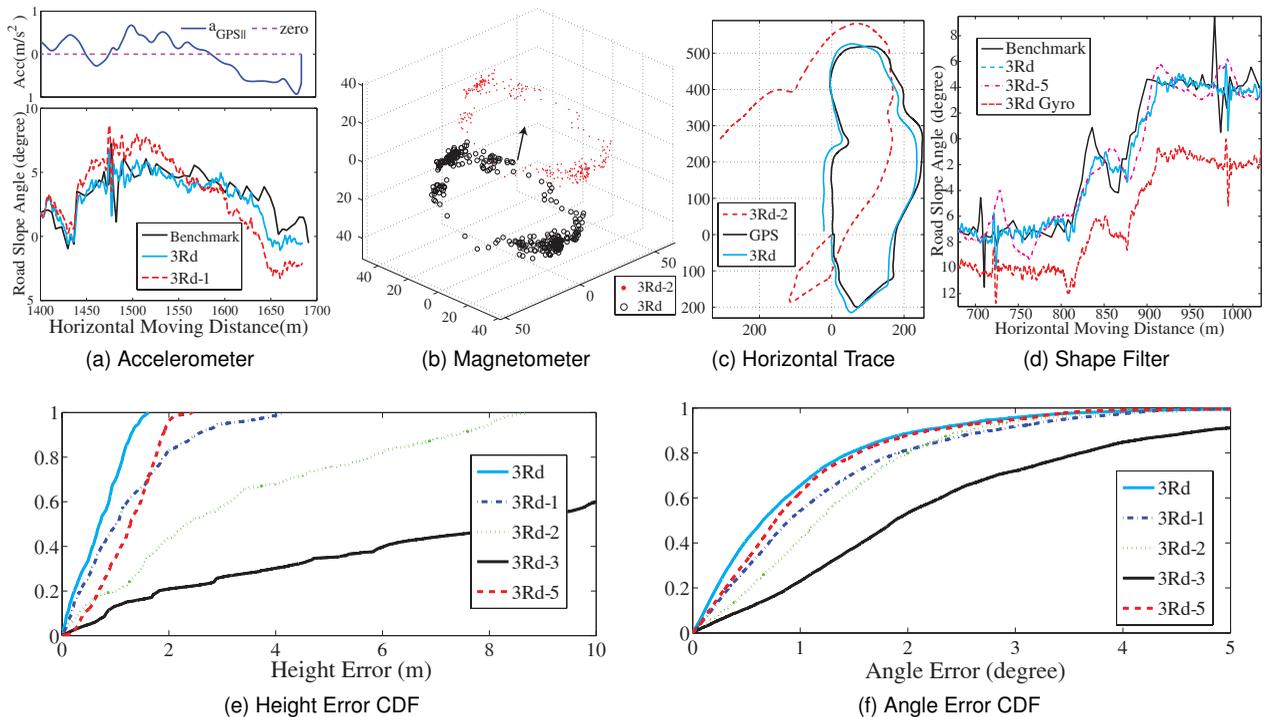


Fig. 13. Components Evaluation

These CDF verify our claim in §3.2.3 that readings of magnetometer decrease the modeling accuracy.

Curve tagged *3Rd-5* in Fig.13(d) is the result without combining the high frequency part of gyroscope measurement. *3Rd Gyro* is the result with full frequency of gyroscope measurement and is obviously biased from benchmark. However, landmarks such as artificial deceleration zones, can be distinguished in the *3Rd Gyro* rather than *3Rd-5*. The improvement of *3Rd* in 13(e)(f) is contributed by the high frequency part of *3Rd Gyro*. Nevertheless, the improvement is limited by the reason that the benchmark is not well aligned with the car measurement in locations due to GPS error. In Fig.13(d), peaks in *3Rd* do not compensate peaks in benchmark even though they are very close. We will discuss this issue in §6

### 5.3 Other Modeling Approaches

#### 5.3.1 GPS Altitude

GPS altitude is an approach to measure the road height. However, reasons such as short receiving antenna make the localization accuracy in mobile phone worser than the official report [16]. A recent study [17] on iPhone3GS shows the median vertical error is about 11m and the median horizontal error is about 9m in ideal outdoor conditions. We get similar result in our evaluation in Fig.11, where we plot the relative altitude and tag it as *GPS*.

#### 5.3.2 Aerial and Satellite Survey

Aerial images are widely used to generate terrain elevation. The National Elevation Dataset [18](NED)

is the primary elevation data product of the USGS. It is available in the U.S. at different resolutions from 30m to 3m. The resolution defined in  $X_m$  means the terrain is measured in unit of grid with side length of  $X_m$ . The mean relative error reported by [18] is 1.64m. For the entire world, a 90m resolution NASA database (SRTM [4]) is available. The 90% relative height error in the Eurasia is 8.7m [4]. We compare our method with SRTM because our experimental area is out of the cover of NED.

The curve tagged *Google* in Fig.11(b) is the road elevation obtained from the Google Earth which integrates SRTM data. The large bump from 800m to 900m is unusual. The reason is that the aerial images are not able to distinguish the terrain elevation the road elevation. This road is dug into a trench rather than floating on the hill. This limitation is universal and it is an obstacle for the aerial survey in supplying accurate road model. Besides these, the resolution is blurred by the grid-based measurement. Any points in one grid share the same elevation and can not provide high resolution road slope as our model.

#### 5.3.3 Barometer Altimeter

Barometer altimeter is an altitude measurement device based on barometric pressure. We compare our results with data collected by Samsung Galaxy S III which has embedded barometer. The input parameters, sea level pressure and temperature, are requested from public database [19] in real time. In Fig.11, Results tagged *Baro* show the barometer

altimeter achieves the best height measurement accuracy among legacy methods. Resent work [20] even uses it as the elevation benchmark.

However, barometer does not fulfill the accurate road modeling for three reasons. First, it is a height measurement tool without directly providing road angle information. Differentiating on smoothed barometer measurement is a direct approach to obtain the slope angle, but the smoothing parameter is normally hard to obtain and the smoothing process reduces the amplitude of slope angle. *Baro* in Fig.11(c) shows the best signal we obtained by smoothing; obvious randomness dominates slope details. This property does not satisfy our slope angle accuracy requirement in §3.1.3 and the impact on real example is shown in §5.4. Second, barometer altitude is sensitive to environmental factors such as the open and close of the car window, air conditioner temperature and so on [21]. Third, barometer is still a rare sensor. To our knowledge, only five high-end mobile phone models are equipped with MEMS barometer.

## 5.4 Application Example

Accurate modeling property, especially the angle variation in *3Rd*, potentially enables new applications. Take the motivating example in the first paragraph of §1. we show how the *3Rd* model helps drivers detect slippery road slope in bad weather.

The maximum friction coefficient between ice and tire is about 0.1 [22], which means the maximum slope angle resisting to slipping on a road covered by ice is  $\theta_{th} = \arctan(0.1)$ . We take this value as the threshold to detect the possible slippery slopes. In this simple example, the evaluation road is divided into segments with equal length of  $20m$ . One segment is considered as slippery if there are more than  $1m$  slope samples larger than  $\theta_{th}$ . We list the results in Fig.14. The horizontal axis is the index of the road segments. *Pos* indicates the slippery segments under our definition. *3Rd* has  $5/86 = 5.8\%$  mismatch when compared with the results of benchmark. However, when we carefully examine those mismatched points, we find 4 of them are caused by GPS location misalignment and they do not affect real use at all. The only false positive one is shown by the arrow. Therefore *3Rd* achieves  $1/86 = 1.2\%$  false positive and  $0\%$  false negative detection in this example. The best known rival 3D model obtained by barometer §5.3.3 achieves  $23/86 = 26.7\%$  in the detection.

## 6 DISCUSSION

We first identify the error sources in our modeling method and then we discuss several practical issues:

1. Frame transformation matrix  $C_b^n(t)$  calculation error. If we treat the moving direction as an known parameter, the slope angle is determined by  $C_b^n(t)$ . As we summarize in §3.2.4, the angle value mainly

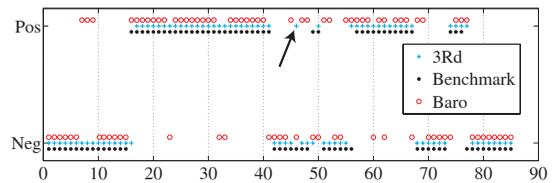


Fig. 14. Slippery Road Slope Detection

depends on the accelerometer direction measurement. Its error accumulates in the height calculation due to the integration property. Even the expectation of height error is zero, its variance increases with moving distance.

2. GPS horizontal error. The locations of the deceleration zones in Fig. 11(d) are shifted from the benchmark. One reason is that the sampling rate of GPS is fixed to 1 second which is too low to get the accurate location when the car is moving. However, the problem is not that obvious when considering traces from multiple runs. The accuracy of the location of the steep slopes can be increased by averaging. The trace level combining/averaging can also be implemented by using the Dynamic Time Warping Algorithm.

3. Moving direction  $\vec{d}^b$  error. The height error is proportional to the product of the horizontal moving distance and the tangent value of the angle error in  $\vec{d}^b$ . The value selected in the learning step may not be the fittest vector in the regular measurement. In Fig.11(b), *3Rd* is rotated for a tiny angle from the optimal value (height at the end should be zero). However, *3Rd* can automatically calibrate the direction value when encountering a loop. Even in a long traveling that does not have a loop path, the result of *3Rd* can be calibrated by other altitude measurement methods. For example, the GPS has large error in the altitude measurement, but in a long range, its error will not increase. These inaccurate estimations can be used as constraints to adjust the value of  $\vec{d}^b$ .

4. Communication and battery overhead. The power consumption of our application is not considered as a problem like other mobile applications. This is because most cars are equipped with a 12V charging port or even USB ports in the vehicle media panel. The communication overhead is also negligible since we do not target real-time road modeling. We note that the rate of sensing data is about 5k Byte/s. Therefore, even after a long drive of several hours or one day, the amount of data is limited to 100MB level, which is quite affordable for current smart devices. Users may upload data whenever they encounter convenient accessible networks.

## 7 RELATED WORK

Using sensors to navigate air craft in 3D space has been widely studied in navigation and robotic area [12]. 3D road modeling is an inverse process of 3D

navigation, but navigation algorithms are not applicable to our scenario due to low sensor quality. The study of using low-cost sensor in navigation focuses on 2D scenarios [23], where the inertial sensors are used to aid vehicle positioning when GPS is temporarily unavailable. One big problem in low-cost sensors is their accuracy. Recent work  $A^3$  [24] proposes a method to find the right time to calibrate gyroscope according to the measurement from accelerometer and magnetometer. However, their method cannot be applied in in-car scenario since car body introduces continuous bias in magnetometer measurement [11].

Work directly related to 3D road modeling can be divided into three categories. The first kind equips camera or laser scanner on vehicles to collect road information. [6] uses the a vision based method to detect road slope, but it does not provide exact angle value. The laser scanner [5] is also a classical tool in 3D modeling.  $3Rd$  is not based on optical method and does not require special equipments. The second kind is based on vehicle sensing. [2] uses mass effect on heavy-duty vehicles to obtain road slope angle. Their method is not scalable to normal vehicles. [1] uses additional vehicle sensors to calculate road slope in real time.  $3Rd$  only needs a mobile phone. Third, aerial images such as SRTM [4] can also be used to get road height information.  $3Rd$  additionally provides accurate slope angle information in the model.

Some work uses mobile systems to sense road conditions. Most of them differ from us by focusing on the traffic condition or driving behavior [25] rather than the detailed road shape. The most relevant one, Pothole Patrol [26] uses the abnormal readings of a mobile sensing platform to detect potholes when the car goes over a pothole.  $3Rd$  aims at providing detailed road shape rather than identifying special road patterns such as potholes or bumps. In [20], they use USGS data to provide accurate elevation for cyclist and use barometer as benchmark. As we show in §5.3.3, barometer is not sufficient to fulfill road angle accuracy requirement.

## 8 CONCLUSION

This paper studies the feasibility of using commodity mobile phone to achieve low-cost and accurate 3D road modeling, which is considered to be a challenging problem due to the low quality sensors and the complex user behavior. However, it is clear that before the real deployment of  $3Rd$ , much more work is required. It would be very interesting to develop crowdsourcing method to improve the model quality among multiple measurement.

## REFERENCES

[1] Y. Sebsadji, S. Glaser, S. Mammar, and J. Dakhllallah, "Road slope and vehicle dynamics estimation," in *American Control Conference*, 2008.

[2] P. Sahlholm and K. Henrik Johansson, "Road grade estimation for look-ahead vehicle control using multiple measurement runs," *Control Engineering Practice*, vol. 18, 2010.

[3] P. Vemulapalli, A. Dean, and S. Brennan, "Pitch based vehicle localization using time series subsequence matching with multi-scale extrema features," in *ACC*, 2011.

[4] "SRTM," <http://goo.gl/cuZYP8>.

[5] A. Jaakkola, J. Hyypya, H. Hyypya, and A. Kukko, "Retrieval algorithms for road surface modelling using laser-based mobile mapping," *Sensors*, vol. 8, 2008.

[6] J. Zhao, M. Whitty, and J. Katupitiya, "Detection of non-flat ground surfaces using v-disparity images," in *IROS*, 2009.

[7] C. Barthold, K. P. Subbu, and R. Dantu, "Evaluation of gyroscope-embedded mobile phones," in *SMC 2011*.

[8] S. Chuazhu, H. Chaohui, and Z. Yong, "Based on adams optimal design of atv motorcycle for gradeability and shock absorption property," in *ICDMA*, 2010.

[9] "Grade warning signs," <http://goo.gl/TKHaxy>.

[10] "NGDC geomagnetic calculators," <http://goo.gl/aFKAov>.

[11] Y. Wang, Y. Chen, J. Yang, M. Gruteser, R. Martin, H. Liu, L. Liu, and C. Karatas, "Determining driver phone use by exploiting smartphone integrated sensors," *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[12] D. Titterton, J. Weston, D. H. Titterton, and J. L. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition*. IET, 2004.

[13] C. M. Shakarji, "Least-squares fitting algorithms of the NIST algorithm testing system," *JOURNAL OF RESEARCH-NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY*, vol. 103, 1998.

[14] "Manual on uniform traffic control devices," <http://goo.gl/uUmk2n>.

[15] "Geometric design of highways and streets," <http://goo.gl/lvgN7f>.

[16] "GPS accuracy," <http://goo.gl/ID645j>.

[17] P. A. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning," *Transactions in GIS*, 2009.

[18] "NED," <http://goo.gl/Uj3Ywm>.

[19] "Hong kong weather," <http://goo.gl/9NtjH7>.

[20] A. Zhan, M. Chang, Y. Chen, and A. Terzis, "Accurate caloric expenditure of bicyclists using cellphones," ser. *SenSys '12*. ACM.

[21] J. Parviainen, J. Kantola, and J. Collin, "Differential barometry in personal navigation," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, 2008.

[22] F. Gustafsson, "Slip-based tire-road friction estimation," *Automatica*, vol. 33, pp. 1087–1099, 1997.

[23] S. Godha, "Performance evaluation of low cost mems-based imu integrated with gps for land vehicle navigation application," *Department of Geomatics Engineering*, 2006.

[24] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," ser. *MobiCom '14*.

[25] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *ACM Sensys*, 2008.

[26] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *ACM MobiSys*, 2008.