SmartScanner: Know More in Walls with Your Smartphone!

Yongpan Zou, *Student Member, IEEE,* Guanhua Wang, *Student Memeber, IEEE,*
Kaishun Wu, *Memeber, IEEE,* and Lionel M. Ni, *Fellow, IEEE*

Abstract—Seeing through walls and knowing clearly what exist inside just like a superman are not only fantastic wishes for humans, but also of much practical significance. For example, you would like to know whether there are pipes, or rebars inside a wall before drilling into it. Moreover, knowing how pipes are configured in a wall before attempting to fix defects would definitely prevent unnecessary damages. Existing methods that intend to address this issue are either costly due to the use of high-end technology, or restrictive for reasons of some strong assumptions. However, in this paper, we present a novel system, SmartScanner, which is based on off-the-shelf sensors embedded in a smartphone. SmartScanner makes full use of in-built sensors, namely, the accelerometer, gyroscope, and magnetometer to achieve this goal inexpensively and conveniently. Specifically, by combining these sensors, we are able to clearly distinguish certain objects inside a wall and map out the layout of an in-wall pipeline system. We implement SmartScanner on two smartphone platforms, namely iPhone 4 and Xiaomi Mi2S, and conduct extensive experiments to evaluate its performance. Experiments show that SmartScanner can achieve high accuracies in distinguishing objects in various scenarios. Meanwhile, as for layout mapping, 90% of length errors are limited to several centimeters for horizontal and vertical pipeline segments, respectively. Also, SmartScanner can achieve centimeter-level position errors of turning points in horizontal and vertical directions in the testbed.

Index Terms—IMU, layout mapping, objects distinguishing

1 Introduction

OBTAINING a perspective of a wall is not just an adolescent fantasy inspired by the movie *Superman*. More importantly, it has great practical significance as well. Nowadays, as space efficiency and aesthetics are increasingly taken into consideration for interior design, in-wall pipeline systems for water supply and draining have become continually popular [1]. What is more, modern architects prefer reinforced concrete structures for fastness. However, inconvenience even dangers can arise when drilling through a wall for practical purposes like repairing defective pipes due to existing in-wall structures. Hitting a rebar while drilling into a wall can destroy the drill itself and severely weaken the concrete structure. In the case where an in-wall layout is unknown, repairing pipes even with just a few defects can become destructive and labour-intensive work [2]. All these add up to absolute necessity of gaining a view of in-wall structures.

Previous work aims at achieving the goal mentioned above can mainly be divided into two parts–distinguishing hidden objects, and mapping hidden systems' layouts, especially pipeline systems. For the first part, some high-end technologies like Ground-penetrating radar (GPR) [3] and multi-sensor technology are adopted whether in research work [4] or commercial products [5] [6], at the price of discommodious access and high cost. As for hidden layout mapping, previous work either holds some strong assumptions which limits its practical use [7], or makes use of sophisticated equipment which makes them not so easily available [4] [8]. However, this paper presents SmartScanner, a smartphone-based application that achieves the goals of detecting and distinguishing certain objects hidden inside a wall, and further mapping a pipeline layout. By taking advantage of off-the-shelf smartphones, SmartScanner avoids utilizing specialized tools, and has relatively high scalability as well.

SmartScanner is based on the fact that ferromagnetic materials like irons, steels and iron-containing alloies can distort a local geomagnetic field and cause local magnetic anomalies. As a result, when a magnetometer is moving close to such kind of objects, any abnormal responses induced by local magnetic anomalies can be detected. What is more, local magnetic anomalies induced by a kind of ferromagnetic object is closely related to the object itself [9] [10] [11], which can be used to differentiate in-wall objects [12] [13].

Specifically, SmartScanner works in two main modes: distinguishing in-wall objects and mapping pipeline layout. In the former mode, a smartphone scans across a wall and detects the magnetic anomaly if an object exists. At the same time, by making use of displacement information acquired by motion sensors (i.e., accelerometer and gyroscope), unique profiles of local magnetic anomaly can be built for the detected objects, which can then be used for distinguishing them. In the mapping pipeline layout mode, we focus on the pipes hidden in a wall. In a similar way,

- *Y. Zou, G. Wang and L.M. Ni are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, 999077.*
  E-mail: {yzouad, gwangab, ni}@cse.ust.hk
- *K. Wu is with College of Computer Science and Software Engineering, Shenzhen University, Nanhai Ave 3688, Shenzhen, Guangdong, P.R.China,518060.*
  E-mail: wu@szu.edu.cn

Manuscript received April XX, 2015

when the smartphone scans across a wall repeatedly, an increasing number of the points where pipes traverse can be located on the wall by combining the motion sensors and magnetometer. After that, by applying a simple curve reconstruction method, the pipeline layout can be reconstructed automatically by SmartScanner.

The idea is straightforward; nevertheless, several challenges need to be solved in order to implement SmartScanner.

First, how to calculate displacement accurately when moving a smartphone over the wall? Accurate displacement is not only necessary for constructing unique profiles for object-distinguishing purpose, but also indispensable for locating points where pipes traverse for later mapping the layout. Although accelerometer-based methods have been proposed for distance estimation, however, they are not appropriate for our application on account of their relatively high errors.

Second, how to distinguish different objects? It is very intuitive to think that objects can be distinguished by the metric of maximal magnetic intensity. However, a previous study [14] and our simple magnetic survey in an apartment declare it to be unreliable and misleading, since the spatial variance of an indoor magnetic field is large.

Third, how to map the pipeline layout automatically? Even when accurate positions of points where pipes traverse are obtained, it is non-trivial to connect them correctly for mapping the corresponding pipeline layout easily. To address this problem, we propose a curve fitting-based curve reconstruction method, taking two reasonable assumptions in consideration in practical scenarios.

Summary of results: we design and implement SmartScanner on two different smartphone platforms: iPhone 4 and Xiaomi Mi2S. Results of experiments in various settings show that SmartScanner can distinguish four common in-wall objects with average accuracies up to 97.6%, 92.8% and 84.2% under three different depths of 5 cm, 7 cm and 9 cm, respectively. Moreover, 90% length errors are 9.8 cm and 6.8 cm for horizontal and vertical pipe segments, and 90% position errors of turning points in horizontal and vertical directions are 9.1 cm and 6.6 cm, respectively.

To sum up, this paper's main contributions are as follows:

- By introducing a special movement, and carefully calibrating acceleration and velocities, this work accomplishes accurate displacement calculation. Based on that, in-wall objects can be distinguished and located.
- We transform the in-wall pipeline layout mapping problem into a problem of connecting points in a plane correctly, which can be solved by a simple proposed method. By this means, some strong assumptions required by previous work can be relaxed.
- Instead of utilizing additional sensors or any specialized tools, we implement SmartScanner on two smartphone platforms. Experiments conducted in a proof-of-concept testbed show that SmartScanner can distinguish common in-wall objects with high accuracy, and map out the in-wall pipeline layout with low length and position errors.

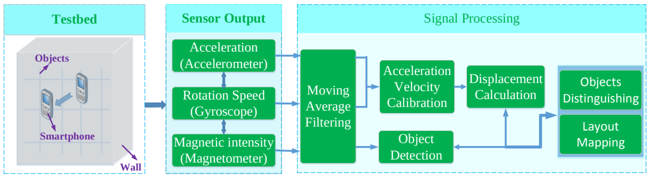


Fig. 1. Framework of SmartScanner

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 gives a concise overview of the SmartScanner. Section 4 presents how to calculate the displacement accurately. Section 5 presents the way to distinguish different objects and locate points that an object pass through. Section 6 describes the experimental setup. Section 7 shows the experimental results and evaluates the performance. Section 9 conducts discussions about SmartScanner. Section 10 concludes this work.

2 Related Work

As a smartphone-based application intending to accmoplish detecting/distinguishing hidden objects and mapping out pipeline system's layout, SmartScanner is closely related to previous work mainly in three aspects: orientation and displacement sensing with an IMU [1]; detecting/distinguishing hidden objects; mapping hidden facilities' layouts.

Orientation and displacement sensing based on IMUs: as various sensors are increasingly equipped in smartphones, sensor-based schemes for indoor positioning and navigation are attracting more and more interest for the sake of energy efficiency, compared with WiFi-based scheme. Among these various schemes, orientation and distance estimations are two critical components, whose accuracies vastly affect the final performance of systems. As a result, great research efforts have been spared on these two aspects. For smartphone orientation estimation, Direction Cosine Matrix (DCM) method is applied for attitude and orientation estimation in [15] [16]. A novel method-DNRF (Drift & Noise Removal Filter) is introduced in [17], in which accelerometer, gyroscope, and magnetometer are fused accurately estimate the orientation of a smartphone whether the phone is stationery or moving linearly. Euler angle and quaternion-based sensor fusion algorithms are developed in [18]. For distance estimation using sensors embedded in a smartphone, a variety of methods are proposed for step counting and step length estimation, thus obtain final estimation of distance. Peak detection algorithm is a common method to detect steps which are utilized in [19]. [20] [21]. A number of step models are proposed for step length estimation based on acceleration [22], walking frequency [23] and a combination of them [24]. By combining step detection and step length estimation, walikng distance can be estimated at different accuracy.

Mapping hidden pipeline layout: As for mapping the layout of hidden utilities, Mapping the Underworld [4] is aimed at mapping out underground infrastructures by

[1] IMU, short for inertial measurement unit, is an electronic device that reports an object's motion states by measuring its accelerations, rotational speeds along with geomagnetic intensity, using a combination of accelerometer, gyroscope and magnetometer.

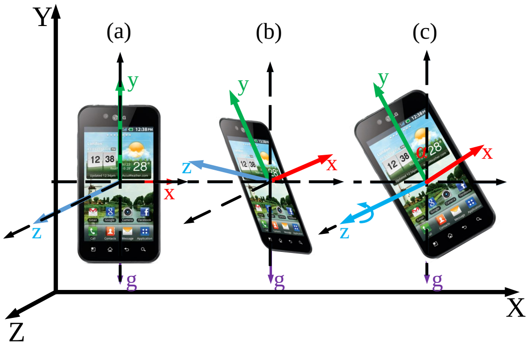


Fig. 2. Rotations of a smartphone in 3D space

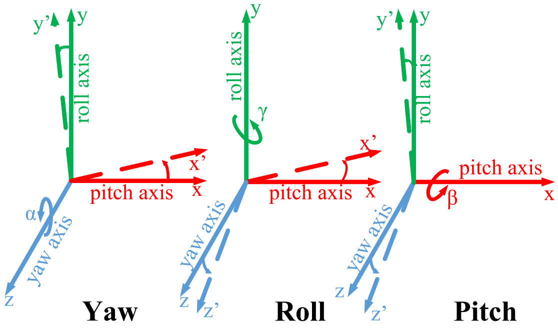


Fig. 3. Illustration of pitch, yaw and roll of a smartphone

means of electromagnetic induction and UWB radar technology. The Pipeline Pigging method is introduced in detail in [8]. In this method, multiple sensors are mounted on a mechanical pig. As the "pig" walks through the whole pipeline system, data of pipelines such as lengths, locations of intersections and defects are collected for later analysis. Although they can achieve relatively comprehensive and high-precision information about pipelines, the specialized tools and sophisticated system designs they employ tremendously hinder their household application. By comparison, SmartScanner does not make use of any specialized tools and sophisticated system design which dramatically reducing the cost and increasing the portability, and can achieve considerable performance for household applications. Another work closely related to SmartScanner is [7], which also aims at mapping out the in-wall pipeline layout. In contrast, [7] utilizes a tiny mobile sensor capsule in which a pressure sensor and a gyroscope are enclosed, to traverse the whole pipeline and collect data to construct the topology of the pipeline system. However, this method has several restricted conditions: the diameter of pipes should be constant and big enough; also, since large amounts of possible paths from entrances to exits exist in a given pipeline system, repeated trials are inevitable. These requirements limit its application in practical scenarios.

Detecting/Distinguishing hidden objects: As for detecting/distinguishing hidden objects, different technologies based on various principles are proposed. Commercial products like metal detectors and wall scanners [5] [6] are available for household applications such as detecting electrical wires or metal objects inside a wall before hammering in a nail or drilling into the wall. The methods adopted by these commercial products usually involve radar technology, which detects and distinguishes objects by analyzing reflected signals. However, differences between these products and SmartScanner are conspicuous mainly in two aspects. First, commercial products employ expensive and specialized tools to achieve accurate distinguishing ability, and thus rendering them not as economical and easily available as SmartScanner. Also, commercial products do not involve mapping of in-wall pipeline layouts, while SmartScanner can accomplish this easily by the combination of motion sensors and magnetometer in a smartphone.

3 System Overview

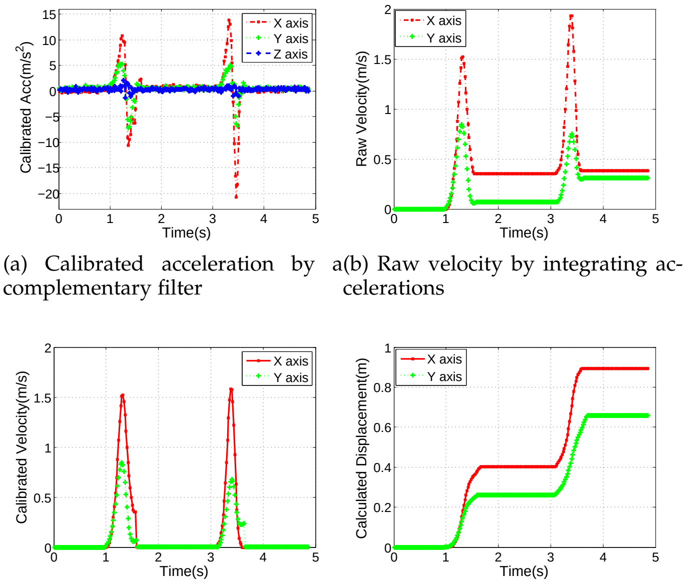
Fig.1 shows the framework of SmartScanner. When a smartphone equipped with a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer scans across the wall, linear accelerations, rotational speed and magnetic intensity can be obtained from the accelerometer, gyroscope and magnetometer respectively. As a preprocessing step, a moving average filter is applied for the above three kinds of readings, in order to remove background noises. After that, accelerations and rotational speed of the smartphone are carefully calibrated and combined such that displacement can be accurately calculated. The calibration process includes dealing with hand rotation and velocity drift. When displacement can be accurately obtained, on one hand, it is combined with magnetometer readings to build profiles of different objects to distinguish them. On the other hand, when a pipeline system exist in a wall, the combination of displacement information and magnetic readings can be utilized to map the pipeline layout.

4 Displacement Calculating

For SmartScanner, accurate displacement is the basis of objects distinguishing and layout mapping. Theoretically, for smartphone-based applications, displacements can be calculated by integrating accelerations twice. However, in practice, errors in accelerations will be accumulated twice, which eventually results in ugly displacement errors. This is the key challenge we shall deal with in this section.

4.1 Strokes Introduction and Data Preprocessing

The first measure we take to fight with accumulative errors is introducing a special movement–stroke. In the context of our application, a stroke is a short period of translational movement beginning and ending in stationary states. The rationale of this special movement is two-fold. First, the relatively high accelerations of strokes are beneficial for suppress errors caused by noises. Second, the relatively short duration of a stroke introduces less interference. Besides, in order to calibrate accelerations and velocities timely, we intentionally set short pauses between consecutive strokes, which can effectively reduce accumulative errors for multiple strokes. As a preprocessing step to remove random interference, we apply a Simple Moving Average (SAM) filter with a span of 7, 4 and 4 on the raw accelerations, gyroscope readings and magnetometer readings, respectively.



(a) Calibrated acceleration by complementary filter

(b) Raw velocity by integrating accelerations

(c) Final processed velocity in X and Y directions

(d) Final calculated displacement in X and Y directions

Fig. 4. The signal processing procedure of displacement calculation

4.2 Dealing with Hand Rotation

Another problem to be handled is hand rotation when moving a smartphone, which causes an additive gravity component projected on each axis. Consequently, obtained accelerations not only contain actual accelerations of strokes, but also incorporate gravity components, as shown in Fig.2. To remove gravity components in accelerations, it is required to estimate rotational angles of the smartphone, denoted by α, β and γ as shown in Fig.3.

To this end, we propose to apply a complementary filter on cleaned accelerations and gyroscope readings, which estimates orientations by skillfully combining them by Eq.1.

$$\theta = \lambda \times \theta_{acc} + (1-\lambda) \times \theta_{gyo} \tag{1}$$

where θ is the final estimation of rotation angles, i.e., α, β and γ. λ represents a filter coefficient, which essentially indicates weights for accelerometer.

θ_{acc} and θ_{gyo} are angle estimations from accelerations and gyroscope readings, respectively. For SmartScanner, since a smartphone is moving against a wall, we assume rotations only happen around the Z axis, that is, only yaw is considered. As a result, the rotational angle (i.e., α) can be obtained from accelerometer and gyroscope by Eq.2, respectively:

$$\alpha_{acc} = arctan(\frac{a_x}{a_y}), \alpha_{gyo} = \int g_z(t)dt \tag{2}$$

where, a_x, a_y are the accelerometer readings in x and y direction. $g_z(t)$ represents gyroscope readings.

Another parameter to be determined is the filter coefficient λ. When a smartphone is held stationary, angle estimation by accelerometer is more reliable, thus the weight for accelerometer should be larger. As a result, a method for identifying the status of a smartphone is required. To this end, we adopt an Exponential Moving Average (EMA) which is defined by:

$$EMA_i = \frac{\sum_{k=i}^{i+N-1} (1-\mu)^{k-i} \times \sqrt{a_{xk}^2 + a_{yk}^2}}{\sum_{k=i}^{i+N-1} (1-\mu)^{k-i}} \tag{3}$$

where μ is a constant equaling to 0.1 in SmartScanner, a_{xi} and a_{y_i} are the i^{th} acceleration samples in x and y directions, and N is the number of samples in sliding windows. When EMA_i is larger than a threshold ε, then a_i indicates the beginning of a stroke and subsequent values are concatenated until EMA_i is less than ε. By identifying the kinestate of the smartphone, appropriate λ can be assigned correspondingly, 0.98 for static status and 0.02 for strokes in SmartScanner.

After obtaining rotational angle α of the smartphone, it is easy to calibrate accelerations by removing projected gravity components on x and y axes. Fig.4a shows the calibrated acceleration by subtracting the gravity components in x and y axes.

4.3 Dealing with Velocity Drift

Intuitively, we can compute velocity by integrating the calibrated accelerations and get the results shown in Fig.4b. Ideally, the velocities of a stroke should be start from and end at zeros, which is determined by characteristic of a stroke. However, it is obvious that drifts exist in above velocities, which brings about large errors in the final displacement calculation.

To mitigate integration errors, we change the strategy of performing integration on accelerations. Specifically, we don't do integration until we detect a stroke using the metric Exponential Moving Average (EMA) as described in Section 3. When the smartphone is detected to be going through a stroke, integration is conducted on the corresponding accelerations to obtain velocity; otherwise, the velocity is set to be zero, indicating the smartphone is in static. By this means, the velocity drifts can be removed as shown in Fig.4c and final displacement can be calculated by integrating the final processed velocity as shown in Fig.4d.

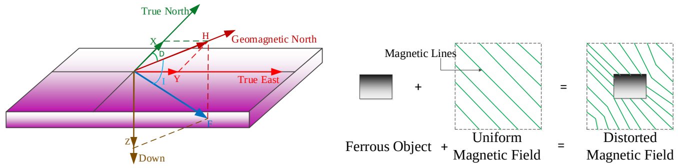
5 Distinguishing Objects

In this section, we will discuss one of the main designing goals of SmartScanner, that is, distinguishing objects. In the beginning of this section, we give a concise introduction of basics of magnetic field and measurements, which consist of preliminaries of objects distinguishing and layout mapping in the following section. Then, we shall give details of the methodology utilized for objects distinguishing.

5.1 Basics of magnetic field

Generally, the term "magnetic field" can refer to a magnetic B field or a magnetic H field. These two fields are distinct but closely related, in vacuum:

$$B = \mu_0 H \tag{4}$$



(a) Description of magnetic field.

(b) Distorted magnetic field caused by ferromagnet

Fig. 5. Magnetic field and distortion

where μ_0 is the magnetic constant ($4\pi \times 10^{-7}$Vs/Am). However, when there is magnetic material, the relationship of these two magnetic fields can be expressed as:

$$B = \mu_0 \mu_r H \tag{5}$$

Where μ_r is the relative permeability of the magnetic material.

As magnetic B field reveals the real cause of the magnetic field, it is more commonly used for describing a magnetic field [25]. The units of B is Tesla(T) or Gauss(G) (1T=10^4G). A magnetic field can be habitually described by seven non-independent parameters: declination(D), inclination (I), horizontal intensity(H), vertical intensity(Z), the north(X) and east(Y) components of the H, and total intensity(F), as shown in Fig.5a. The total intensity of the geomagnetic field at the earth's surface varies from 23 Gauss to 66 Gauss [26].

Magnetic field intensity can be measured by two types of magnetometers: vector sensors and total field sensors [27]. The former measures a single component of the magnetic induction. That is, for a vector magnetometer, B_x, B_y, and B_z are all obtained. However, for the latter, it only measures the magnitude of the field without its direction. The total field intensity can be obtained from the vector magnetometer:

$$B_{total} = \sqrt{B_x^2 + B_y^2 + B_z^2} \tag{6}$$

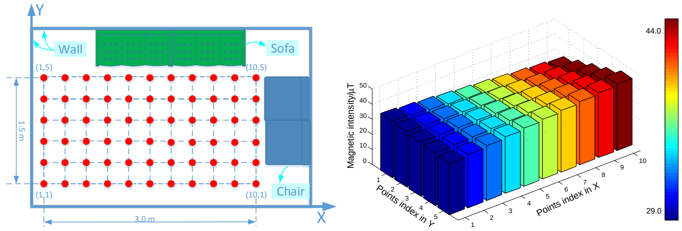
Where B_x, B_y, B_z are magnetic intensity component along X, Y and Z axis in Fig.5a.

Earth's undistorted magnetic field in a small-scale region will be of a constant magnitude and direction, since the magnetic gradient of the geomagnetic field is 10nT/km. However, when a ferrous object like an alloy steel, a cast iron, or a steel reinforcement of panel is present in the geomagnetic field, the object will become magnetized by the primary magnetic field (i.e., geomagnetic field) and induce a secondary magnetic field. The resulting field is a combination of the primary and secondary fields. As a result, the local magnetic field will be distorted as shown in Fig.5b, and consequently creates magnetic anomalies in the resultant magnetic field which can be detected by magnetometer. Based on the anomalies detected by a magnetometer, hidden objects can be detected.

5.2 Distinguishing objects

5.2.1 Constructing magnetic profiles

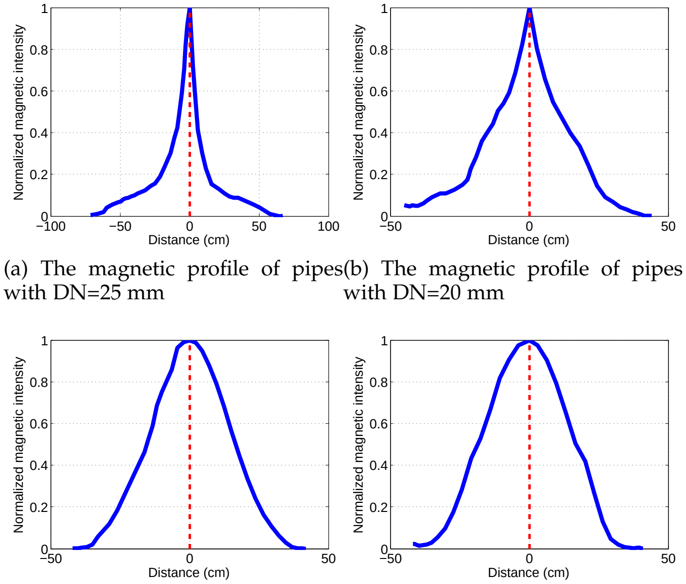
To distinguish in-wall ferromagnetic objects by only a smartphone, a straightforward idea is to compare the maximal magnetic anomalies caused by objects, since intuitively objects with stronger magnetism cause larger magnetic anomalies.



(a) Magnetic survey conducted in the apartment

(b) Spatial variance of magnetic field in surveyed area

Fig. 6. Magnetic survey in an apartment and the corresponding result



(a) The magnetic profile of pipes with DN=25 mm

(b) The magnetic profile of pipes with DN=20 mm

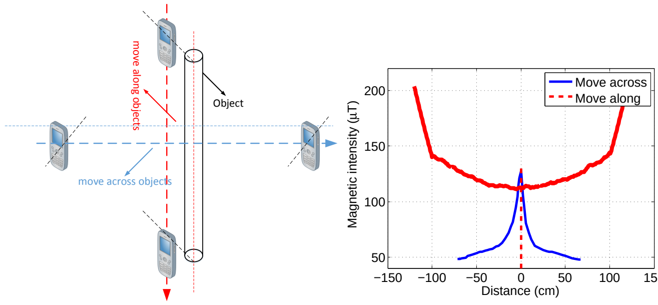
(c) The magnetic profile of rebars with DN=10 mm

(d) The magnetic profile of rods with DN=7 mm

Fig. 7. Magnetic profiles of tested objects

However, this simple idea does not work well practically. The primary reason is that intensities in an indoor magnetic field vary significantly due to steel shells in most modern buildings. As a result, magnetic anomalies caused by even the same object may not be the same in different positions. The spatial variance of indoor magnetic field is reported in [14] and verified by our survey conducted in an area of 3.0 m×1.5 m. The surveyed area is meshed with grids as shown in Fig.6a. We record magnetic intensities in positions denoted by red points using a smartphone for a period of time and then average the recordings to remove magnetic variance in time domain. The results are shown in Fig.6b, from which we can clearly see that the maximum difference of magnetic intensity is up to 13 μT. This noticeable magnetic variance is comparable with magnetic difference of some objects, and consequently causes difficulties for distinguishing objects using a single intensity as the metric.

Our idea of solving this problem is inspired by magnetic field theory [28]. When a magnetometer crosses over a certain ferromagnetic object, magnetic anomalies detected vary with the distance between the object and magnetometer. More importantly, different objects (of different materials, or sizes) present different relations between magnetic anoma-



(a) Sketch of the verification experiment (b) Results of the verification experiment

Fig. 8. Sketch and results of the verification experiment

lies and displacements, which can be viewed as unique profiles of objects for classification purpose as reported in [29].

The distance between an object and a smartphone in our application is defined as the distance between axial center of the object and magnetometer embedded in the smartphone. This distance can be acquired when moving a smartphone by methods described in Section 4. As a result, it is a relatively easy task to construct magnetic profiles by simply combining displacements calculated from motion sensors with magnetic anomalies detected by a magnetometer. Moreover, in order to reduce the influence of spatial variance of magnetic intensities, we normalize the obtained magnetic intensities by:

$$\tilde{y} = \frac{y - min(y)}{max(y) - min(y)} \tag{7}$$

where $\tilde{y}$ and y represent normalized and absolute magnetic intensities, respectively. Fig.7 show magnetic profiles of objects which commonly exist in walls and utilized for our experiments.

Practically, it is noted that magnetic profiles of objects should be constructed when a smartphone move across objects, instead of moving along objects, especially for long cylindrical objects. The main reason is that magnetic anomalies across long-cylindrical objects are more obvious than those along objects within a range of tens of centimeters, except at both ends. We conduct an experiment to verify this, in which we measure intensities of magnetic anomalies along and across a galvanized steel pipe as shown in Fig.8a. The pipe has a nominal diameter of 25 mm and is placed behind a wall with a thickness of 7 mm. The result is displayed in Fig.8b. As we can see, the magnetic anomalies along the pipe present a flat trend in a large distance range.

As a result, for practical use of SmartScanner, we need to first detect objects when we move a smartphone over a wall, which can be easily achieved by tracking the average magnetic intensities within a sliding window during scanning across the wall. When the average values encounter a sharp change, an unknown object is declared to be detected in the wall. After that, we move the smartphone around the object in horizontal and vertical directions to find out the direction of moving across the object.

5.2.2 Training SmartScanner

Constructing magnetic profiles of objects is a prerequisite for system training stage. In this stage, we place each of the L objects (each object is attached with a class label among $1, 2, ..., K$) behind a wall of a certain thickness and then move a smartphone on the other side of the wall for M times to construct candidate profiles. In the following, we perform calibration on those candidate profiles of each object, which is aimed at means picking out a subset of exemplars among all candidates. To accomplish this, we borrow the concept of proximity-based outlier detection in pattern recognition, in which an object is regarded to be an outlier if it is far away from the remaining objects. Specifically, for each object, we construct a $M \times M$ proximity matrix $A_{M\times M}$, in which element $A_{i,j}$ represents the proximity between i^{th} and j^{th} candidate profiles defined in Section 5.2.3. The final proximity score of each candidate profile is defined as the summation of its proximity values to other $M-1$ candidates. Finally, we pick out and store the top N candidates with lowest proximity scores as our final exemplars used for objects distinguishing.

5.2.3 Distinguishing objects

When we detect an object in a wall and construct its magnetic profile using the method mentioned in Section 5.2.1, it is then required to perform profile matching with the pre-stored profiles of certain objects. Before that, a metric for measuring the similarity between any two profiles P_i and P_j needs to be defined. We adopt Normalized Euclidean Distance (NED) which is defined as follows for this purpose:

$$NED(P_i, P_j) = \frac{\sqrt{\sum\limits_{k=1}^{N} (P_i(k) - P_j(k))^2}}{\sqrt{\sum\limits_{k=1}^{N} P_i(k)^2} \times \sqrt{\sum\limits_{k=1}^{N} P_j(k)^2}} \tag{8}$$

where $P_i(k)$ and $P_j(k)$ are the k^{th} sample in P_i and P_j, respectively. N is the total number of samples in P_i or P_j. Obviously, the smaller $NED(P_i, P_j)$ is, the more similar P_i and P_j are. Another thing about this metric needs to be noted is that for this similarity metic, equal number of samples are required for two profiles. However, this cannot be always satisfied in practical scenarios as sampling time and rate may be different. This issue can be handled by linear interpolation [30].

Based on the above metric, we can calculate similarities between the unknown profile and all $L \times N$ pre-stored profiles and consequently get $L \times N$ NEDs attached with class labels. Next, we adopt a distance-weighted voting scheme for making the final decision. To be specific, among the $L \times N$ NEDs, we select the top k nearest neighbours, that is, the top k ($k \geq L$) smallest values denoted by $d_1 \leq d_2 \leq ... \leq d_k$, with each labeled by a corresponding object index $l_i \in \{1, 2, ..., L\}$. By assigning weight to each of them as follows:

$$w_{d_i} = \begin{cases} \frac{d_k - d_i}{d_k - d_1}, & d_k \neq d_1 \\ 1, & d_k = d_1 \end{cases} \tag{9}$$

Based on this, the final object distinguishing result can be acquired by the majority weighted voting:

$$l = \arg\max_{l} \sum_{T} w_{d_i} \times \delta(l = l_i) \tag{10}$$

where $T = \{l_1, l_2, ..., l_k\}$, and $\delta(l = l_i)$ represents the Dirac delta function defined as follows:

$$\delta(l = l_i) = \begin{cases} 1, & l = l_i \\ 0, & l \neq l_i \end{cases} \tag{11}$$

Once the index l is determined, the object can be recognized.

6 Mapping Layout

In this part, we focus on another significant functionality of SmartScanner, that is, mapping an in-wall pipeline layout. For layout mapping, it is required to recover an unknown layout with a high accuracy in geometrical size and spatial position. On a high level, the idea is to recover a pipeline layout by locating and connecting points on the wall through which pipes traverse. In the following, we shall detailedly introduce the techniques utilized for mapping a pipeline layout.

6.1 Locating Pipes

As mentioned above, the first step of layout mapping is to locate points on a wall through which pipes traverse. To this end, magnetic anomalies caused by pipes can be utilized since magnetic anomalies of objects present certain distribution in space around the source of anomalies. As a result, by combining displacement calculation and measurement of magnetism, it is feasible to locate the source of magnetic anomalies (pipes in our application).

However, it is unrealistic to utilize modeling method to obtain exact spatial distribution of magnetic anomalies caused by pipes in this paper due to the tremendous complexity [31]. But indicated by the model built for magnetic anomaly detection (MAD) of underground ferromagnetic pipeline in [31], a simple but effective scheme can be applied to locate points where pipes traverse. In this scheme, when we move a smartphone across a pipe and obtain corresponding magnetic profile by combining displacement calculation and magnetic measurements, the axial center of the pipe can be located at directly under the maximum field position, that is, the turning point in magnetic profiles at which the increasing trend of intensities switches to be decreasing, as shown in Fig.7.

A little modification needs to be made when we apply this scheme in practical scenarios, taking magnetic noise in the background into consideration. As a result, we take the average value of distances when corresponding magnetic intensities are within 95% of the maximum. By this method, influence of background noise can be effectively suppressed.

In practice, we move a smartphone across a wall to sample points where pipes traverse in the way as shown in Fig.16. Each time we start from a certain reference point (RP1, RP2, RP3 or RP4) with known exact relative positions and try to move horizontally and vertically stroke by stroke. The reason why we try to maintain horizontal or vertical is that, as we will see later in Section 8.1, errors of displacement calculation when moving a smartphone horizontally or vertically are smaller than those in an inclined way. For the sake of reducing errors in locating pipes, it is better to move in such a way. What's more, since displacement errors of each stroke will be accumulated after multiple strokes, as a result, we divide the whole scanning area into multiple regions, with each containing a reference point. In this way, a long scanning distance is cut off into shorter segments.

6.2 Mapping Layout

As mentioned above, when we move a smartphone across a wall, points where pipes traverse can be located on the wall. When sufficient number of such points are obtained after scanning processes, the layout of an in-wall pipeline system can be mapped by connecting them deliberately in an appropriate way.

Essentially, the layout mapping process is a curve reconstruction problem, which can be defined as: given a set of sample points on unknown curves on a plane, the problem of connecting them according to their spatial adjacencies is referred as a curve reconstruction problem [32] [33]. Due to its significance in various areas, plenty of techniques and algorithms such as minimum spanning tree (MST) [34], α-shape [35], r-regular shapes [36], CRUST [32], NN-CRUST [37], CONSERVATIVE-CRUST [38], have been proposed for different cases considering different aspects of the problem such as whether sample points are sampled uniformly or non-uniformly. However, they are not appropriate for our case where two practical observations for a pipeline layout can be taken into account, considering the complexity of these classical algorithms. The practical observations that can aid the reconstruction process are summarized as follows.

- *Observation 1: Pipe segments consisting of a pipeline system are considered to be straight lines.* It is an obvious observation in our practical scenarios.
- *Observation 2: The length of each pipe segment in a pipeline system should be larger than a certain threshold, say 20 cm in our case.* Actually, this observation is actually based on industry norms of plumbing system, since more conduit joints will be required if pipe segments are too short.

Based on the above observations, we propose an two-stage method specific for our layout mapping purpose. On a high level, we utilize a bidirectional sliding window (named as a sliding box), moving along horizontal and vertical directions, to do curve fitting for points and obtain short line segments within each sliding box. Afterwards, we merge these line segments in sliding boxes along horizontal and vertical directions to recover the whole pipeline system.

Specifically, for the first stage, namely, curve fitting in each sliding box, a straightforward idea is to fit all points directly, forming a single line segment in each sliding box. However, this idea is not favorable when taking two points into considerations: 1) there exist noise in the positions of points in sliding boxes, which will make the fitting result biased; 2) for the case where a sliding box contains intersections as shown in Fig.9, directly fitting all inner points will result in great distortion, since points in such sliding boxes form two clusters and each can be fitted

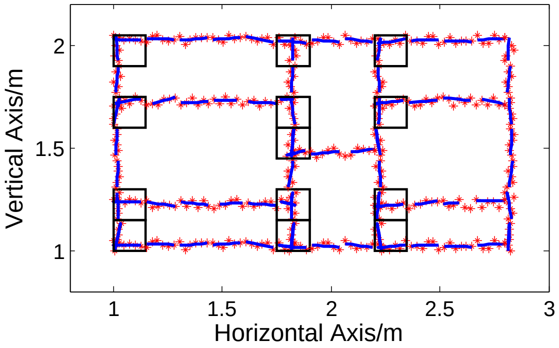


Fig. 9. Sliding boxes containing intersections

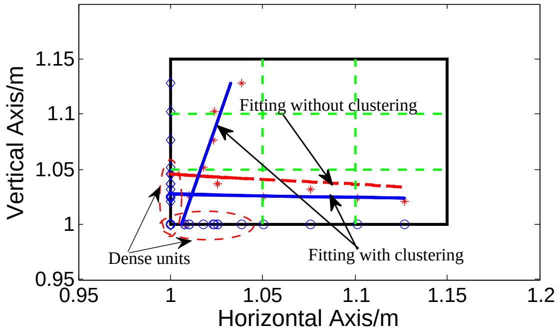


Fig. 10. Curve fitting in sliding boxes

to be a straight line segment (based on *Observation 1*) in different directions. As a result, before conducting curve fitting, we first perform clustering on data points to figure out all the cluster(s) and then apply Linear Least Squares Fit (LLSF) on each data cluster as shown in Fig.10. The clustering technique we applied is something like dense unit-based subspace clustering [39]. To be specific, we mesh each sliding box with grids of certain size and project data points within the sliding box onto X and Y dimensions. In this way, we can find out dense units in both dimensions, which are defined as units containing projected points with densities exceeding a threshold d. Once the dense units in each dimension is determined, the corresponding points projected in the dense unit form a cluster and are fitted to be a straight line using LLSF. Fig.10 shows the results of curve fitting with (blue lines) and without (red line) clustering, respectively. It clearly demonstrates that clustering before curve fitting is beneficial for recovering the actual shape of curves.

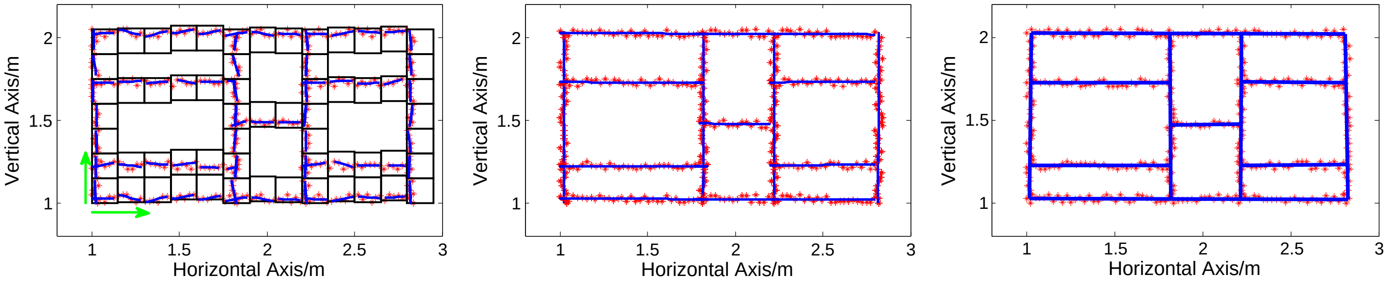
After the previous stage, we can obtain line segments characterized by slopes and intercepts in sliding boxes as shown in Fig.11a and meanwhile corresponding points used for fitting are stored for each sliding box. The data for merging segments are stored in the form of a matrix, in which each element is data for the corresponding box, including slopes and intercepts of line segments and points for curve fitting. In the following, we perform piecewise curve fitting to merge line segments in sliding boxes to recover the whole pipeline system. More specifically, we scan elements in the matrix horizontally and vertically and merge line segments in consecutive sliding boxes by comparing their slopes and intercepts. The merging process is accomplished by adding points and fitting curves progressively, until encountering null elements meaning the end of a line segment. After merging line segments in all sliding boxes, the layout of a whole pipeline system can be recovered as shown in Fig.11b. However, the obtained pipeline layout after merging segments requires to be refined, since there exist ledges and gaps in the present version. For removing ledges and filling gaps, we can simply determine coordinates of intersections and extend pipelines to fill gaps and remove ledges.

It is noted that there are several parameters in the above method to be determined. We give a concise discussion on the determination of these parameters as follows.

- *the size of sliding box:* the size (width and height) of a sliding box is determined empirically based on *Observation 2* and error range of displacement calculation. In practice, the size of a sliding box should be less than the length of the shortest pipe segment and larger than errors of displacement calculation.
- *the density of sampling points:* in our context, the density of sampling points is defined as the number of points per-meter of pipes. The density of sampling points has effect on the performance of layout mapping, since intuitively denser points lead to more favorable mapping performance. As for this parameter, we give a specific evaluation its effect on the performance of layout mapping in Section 9.
- *the density threshold for clustering:* the point density of a unit is defined as the percentage of points contained in each unit, compared to the total number of points in each sliding box. Actually, the density threshold for clustering is determined by grid size and sampling point density. For the case shown in Fig.10 where point density is about 40/m and grid size is 5 cm, the density threshold for clustering in our implementation is $\frac{2}{3}$.

7 EXPERIMENTS SETUP

We implement prototypes of SmartScanner using an iPhone 4 and a Xiaomi Mi2S, both equipped with 9-axis IMUs. Components of this testbed include galvanized steel pipes of different sizes (DN=25 mm and DN=20 mm, denoted by DN25 and DN20 respectively for convenience. DN representing nominal diameter), reinforcing rebars (DN=10 mm) and iron rods (DN=7 mm). All these objects are common in walls of modern buildings for diverse purposes such as plumbing systems, concrete structures and structure supports. Besides, a wall of 9 cm in thickness and slates of 7 cm and 5 cm in thicknesses are used for the experiments. The reason why we use slates for our experiments is that we investigate the influence of buried depth on distinguishing objects hidden in a wall and also consider practical difficulties of finding appropriate testbeds. For convenience of description, the slates are referred to be walls as well in our context. Experiments are carried out in three main sections: 1) experiments to evaluate displacement-calculating performance; 2) experiments to test object-distinguishing accuracy; 3) experiments to examine the layout-mapping performance.



(a) Stage 1: Curve fitting in sliding boxes

(b) Stage 2: Merge Line segments

(c) Stage 3: Refine raw layout

Fig. 11. Three-stage layout mapping process

7.1 Experiments of displacement calculating

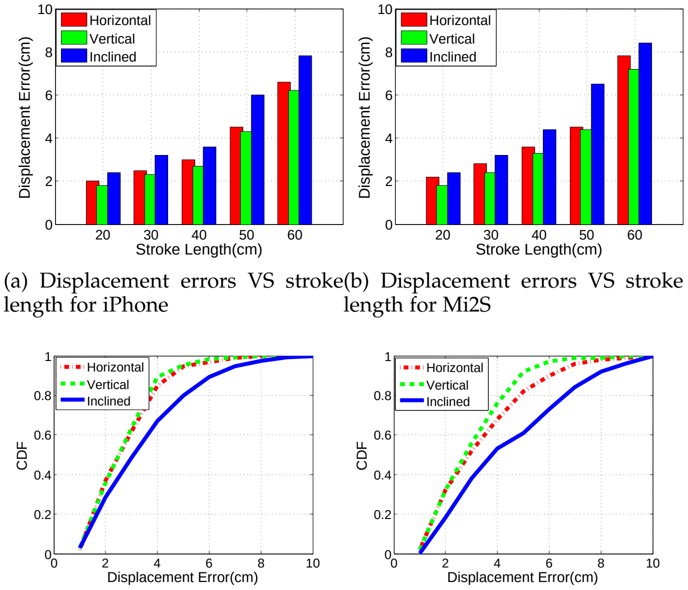
The performance of displacement calculation lays the foundation of SmartScanner for distinguishing objects and mapping layout. As a result, we evaluate it in diverse scenarios in the first section of experiments. To be specific, the performance is evaluated under three different ways of moving a smartphone, namely, horizontally, vertically and obliquely. During experiments, volunteers are firstly required to move a smartphone against a wall in a certain way (horizontally, vertically or obliquely) for various distances (varying from 20 cm to 60 cm) in a single stroke. The error in each case is calculated by taking the average value of repeated experimental results. By this, we can pick out an optimal range of stroke lengths for moving a smartphone each time. What's more, we require volunteers to move a smartphone across a wall at stroke lengths within the selected range and obtain the cumulative distribution function (CDF) of displacement errors of each stroke. In each subpart of experiments, a total number of 10 volunteers are involved and each conducts 20 repeated experiments.

7.2 Experiments of objects distinguishing

In the second section, we test accuracies of distinguishing objects under three different cases. Practically, pipes (DN25 and DN20), rebars and rods are put behind walls of 5 cm, 7 cm and 9 cm in thickness on one side, which simulates cases where objects are buried in walls at different depths. The height of testbeds (walls) can be adjusted from 1.5 m to 2.5 m. In each case, we move a smartphone across objects on the other side of the wall and obtain magnetic profiles of each object as demonstrated in Section 5.2.1. After that, we perform K-fold cross-validation to do training (Section 5.2.2) and classification (Section 5.2.3), and obtain the average classification accuracy. The total number of profiles we obtained for training and classification in each case is 400, with 100 belonging to each object. Critical parameters in training and classification process are listed as follows: K=2 (thus M=50), L=4, N=25 (50%), k=50 (50%). Moreover, in order to examine performances under different numbers of training samples, we vary the number of training samples of each object and gain the corresponding average distinguishing accuracies.

7.3 Experiments of layout mapping

In the third section, a pipeline layout as shown in Fig.16 is used for performance evaluation of layout mapping. In this layout, each line segment represents a galvanized steel pipe (DN=20 mm), which is commonly used in a household plumbing system. The built pipeline system is put against a wall with a median thickness of 7 cm. A smartphone scans against the other side of the wall to locate points through which pipes traverse. Each time a scanning process starts from certain predetermined points (reference points shown in Fig.16) and scans the wall horizontally and vertically stroke by stroke. After multiple times of scanning processes, enough points can be sampled and the layout can be mapped out via the method described in Section 6. We conduct a total number of 200 repeated experiments on the testbed using iPhone 4 and Mi2S platforms. We also vary the number of points used for layout mapping to evaluate the performance under different points densities.



(a) Displacement errors VS stroke length for iPhone

(b) Displacement errors VS stroke length for Mi2S

(c) CDFs of displacement errors for iPhone

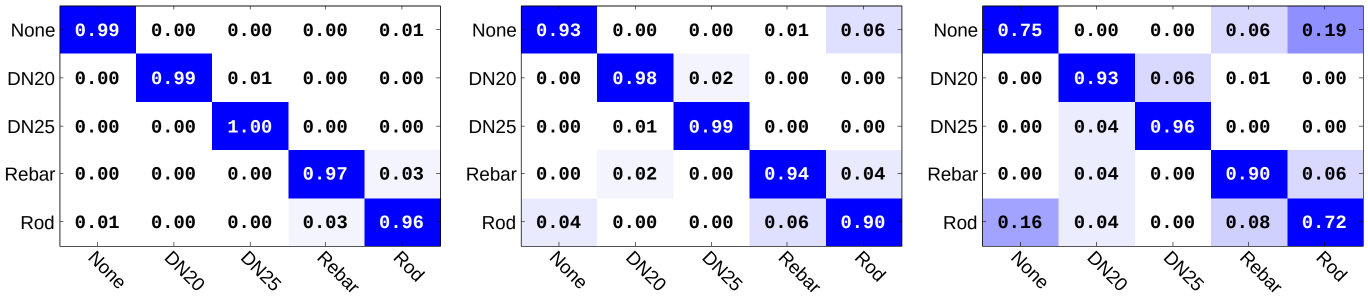
(d) CDFs of displacement errors for Mi2S

Fig. 12. Evaluation of displacement calculation

8 Performance Evaluation

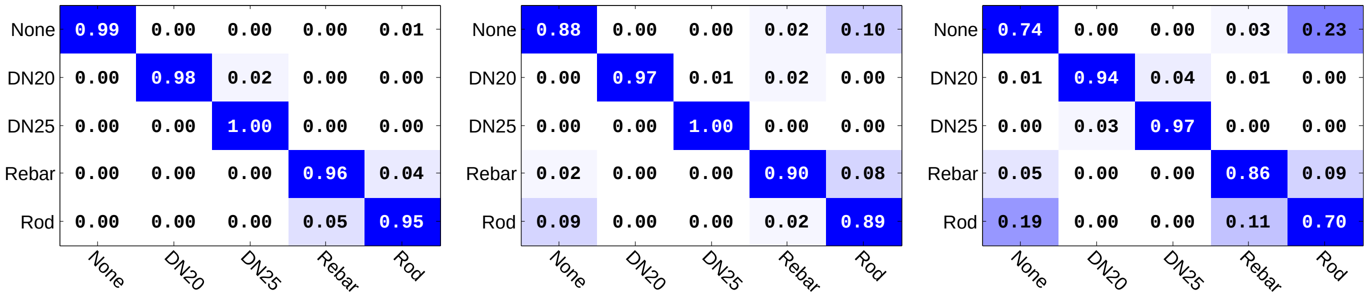
8.1 Displacement-Calculating Accuracy

Fig.12a and Fig.12b plot the displacement calculation errors varying with length of a single stroke for iPhone 4 and



(a) wall thickness=5 cm (b) wall thickness=7 cm (c) wall thickness=9 cm

Fig. 13. iPhone's performance of distinguishing objects under different wall thicknesses



(a) wall thickness=5 cm (b) wall thickness=7 cm (c) wall thickness=9 cm

Fig. 14. Xiaomi's performance of distinguishing objects under different wall thicknesses

Xiaomi Mi2S. As we can see, errors of single strokes increase with their corresponding lengths. The reason is that when the displacement of a stroke increases, it is harder to move a smartphone in a controlled way as required. However, it is noted that although minimum errors can be achieved when the length of a stroke equals 20 cm, it is more preferable to move a smartphone by about 40 cm in each stroke. The reason is that more strokes are required for a fixed long distance when moving a smartphone in shorter-length strokes, which introduces larger cumulative errors. And also, in practice, it is easier and more comfortable to move a smartphone in strokes of such a length. Based on these considerations, in the following experiments, we move the smartphone for about 40 cm in each stroke.

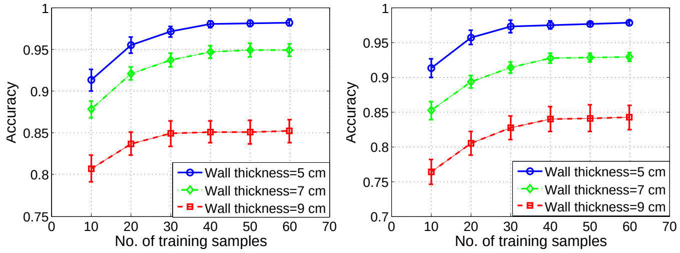
Fig.12c and Fig.12d plot the CDFs of displacement calculating errors of iPhone 4 and Mi2S when users conduct horizontal, vertical and inclined movements in a length of about 40 cm for each stroke. It can be seen that for the above three movements, 90th percentile of errors are 4.2 cm, 4.5 cm and 6.3 cm for iPhone platform. In contrast, for Mi2S platform, 90th percentile of errors are 4.4 cm, 5.9 cm and 8.0 cm respectively. The performance of iPhone platform is better than that of Mi2S platform due to its more superior IMU sensors. The range of displacement errors are critical for determining the size of sliding boxes as described in Section 6.2. Besides, there is another point to be paid attention to that vertical strokes cause less errors than horizontal ones. This is because horizontal strokes usually accompany with more rotations due to wrist movements than vertical ones. Although effect of rotations on accelerations can be eliminated via calibration, this will also lead to some errors.

8.2 Objects-distinguishing Performance

8.2.1 Accuracy of objects-distinguishing

Fig.13 shows the performance of SmartScanner running on iPhone 4 in distinguishing under different wall thicknesses. As we can see, an average accuracy of 98.2%, 94.8%, and 85.2% can be obtained when the thickness of the wall is 5 cm, 7 cm and 9 cm, respectively. Considering the factor of wall thickness, the overall performance will degrade when the wall thickness increases. This is because the thicker a wall is, the less accurate the measurements of magnetic anomalies are. This is particularly obvious for rebars and rods, since the magnetic anomalies caused by them are smaller compared with pipes and are easily interfered by noise. The reason behind this is due to differences in materials and sizes, both of which have effects on magnetic anomalies. Consequently, we can see that it is the easiest to differentiate pipes, yet the hardest to differentiate rods, especially when the wall thickness increases. Another thing to be noted is that the diameter of a pipe will have effect on magnetic profiles such that pipes of different diameters can be distinguished. We will give detailed discussion about this issue in Section 9.2.

Similarly, Fig.14 shows the performance of SmartScanner running on Mi2S. It can be seen that the average accuracies of distinguishing objects under wall thicknesses of 5 cm, 7 cm and 9 cm are 97.6%, 92.8%, 84.2%, respectively, which are comparable with those of iPhone platform. What's more, the overall trend of objects-distinguishing performance is similar to that on iPhone platform. The above results demonstrate the reliability of SmartScanner in displacement calculating, regardless of the platform.



(a) Accuracy VS No. of training samples for iPhone

(b) Accuracy VS No. of training samples for Mi2S

Fig. 15. Training overhead of SmarScanner

8.2.2 *Effect of the number of training samples*

Intuitively, more training samples contribute to better distinguishing accuracy. In order to gain more insights about this, we evaluate this training overhead in this part. Specifically, we vary the number of training samples of each object from 10 to 60. Fig.15a and Fig.15b show the average distinguishing accuracy in each case for iPhone and Mi2S. As indicated by the results, the performance can be increased with an increasing number of training samples, especially when the number of training samples is small. However, when the number of samples is up to about 30∼40, the performance improvement is negligible. As a result, the training overhead of SmartScanner is about 30∼40 for a satisfactory performance.

8.3 Evaluation of Layout Mapping

In this part, we evaluate the performance of layout mapping process from three distinct perspectives, namely, the length accuracy, the position accuracy and the effect of point density.

8.3.1 *Length Accuracy of Layout Mapping*

The first evaluation aspect is the accuracy of geometric dimensioning of the mapped layout. To be specific, we adopt length error as our metric for measuring the performance as done in [7], which is defined as the absolute length difference of a pipe segment in the mapped layout and the original one, as shown in Fig.17. We conducted 200 repeated experiments and calculate the CDFs of length errors of horizontal and vertical pipe segments. Fig.18a shows the results of length errors. As we can see, its 50th percentile is 5.6 cm and 3.9 cm for horizontal and vertical pipe segments respectively, and 90th percentile is 9.8 cm and 6.8 cm respectively. The length accuracy of layout mapping of SmartScanner is comparable with that of Pipeprobe [7] (90th percentile is 10.0 cm and 5.4 cm for horizontal and vertical pipes). The results show that SmartScanner can achieve centimeter-level mapping accuracy in geometric dimensioning, which demonstrates the effectiveness of SmartScanner.

8.3.2 *Position Accuracy of Layout Mapping*

Another metric of performance evaluation is the position accuracy of the mapped layout. Different from the definition of length error, we define the position error of a turning point in the pipeline as its absolute coordinate difference in the mapped layout and the original one, as shown in Fig.17. Similar to the above evaluation, Fig.18b shows the CDFs of position errors of turning points in both horizontal and vertical directions based on 200 experiments. As we can see, the median position errors in horizontal and vertical directions are 6.2 cm and 4.3 cm, respectively. Correspondingly, the 90th percentile of errors are 9.1 cm and 6.6 cm, respectively. As a result, the overall positional error of SmartScanner is slightly better than that of PipeProbe [7], of which the 90th percentile of positional error is roughly 15.8 centimeters. Essentially, position errors of turning points can indicate the spatial distortion, including translation and rotation, of pipe segments in the mapped layout. The results of position errors illustrate that SmartScanner can map the layout with a high spatial accuracy.

9 Discussion

In this section, we try to clarify several issues about SmartScanner, which helps to gain a more complete understanding of this system.

9.1 Issues about displacement calculating

As for removing gravity components in accelerations, in fact, there are some software APPs such as Sensor Kinetics Pro on Android and iOS platforms, which is a well-known paid APP for sensor data collection and contains a derived software sensor–gravity sensor that can output pure gravity component on each axis. However, according to our testing results of Sensor Kinetics Pro on Mi2S, there is one big problem with this gravity sensor, that is, the response rate of this software sensor is not desirable and can lead to greater errors in displacement calculation, compared with the method utilized in SmartScanner. What's more, SmartScanner requires users to move a smartphone in a controlled way, which consequently brings about a certain degree of inconvenience. However, it is noted that according to our experimental results as shown in Fig. 4(c), the maximum velocity of moving a smartphone is about 1.5 m/s and lasts for a rather short period. This means that this special movement shall not cause much discomfort to users in order to mitigate cumulative errors, which is also verified by our user study after experiments.

9.2 Issues about objects distinguishing

In regard to objects distinguished by SmartScanner, it is reasonable to limit them to certain kinds, namely, pipes, rebars, rods, electric wires and cables, since considering our application scenario, that is, household in-wall objects distinguishing. Although practically it is hardly feasible to detect electric wires and cables by common smartphones at present according to our tests, however, we envision that with higher-precision and more advanced magnetometers equipped in smartphones, it is easy to extend the categories of objects that can be distinguished by SmartScanner. What's more, experiments to explore the effect of diameters show that pipes of different diameters have different magnetic profiles and will be distinguished with a high accuracy. As a result, we treat pipes of different diameters as different objects and construct profiles them. This seems to be a bit troublesome due to the great effort to train the system. However,

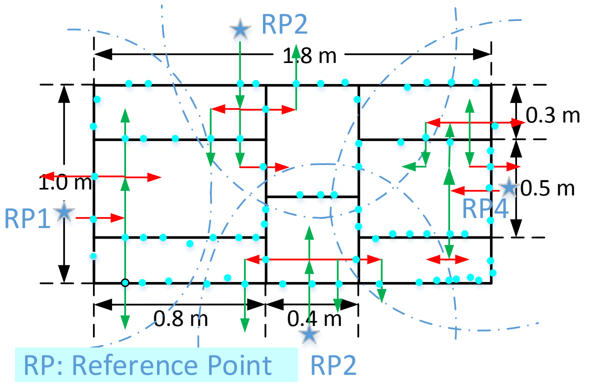


Fig. 16. Pictorial description of sampling points process

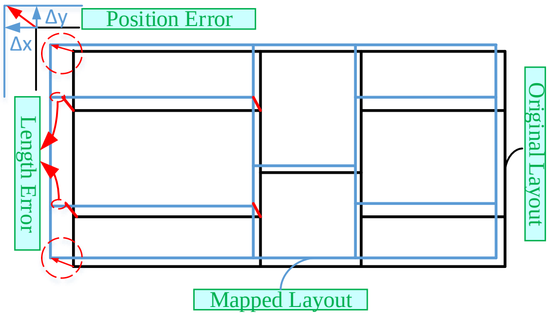
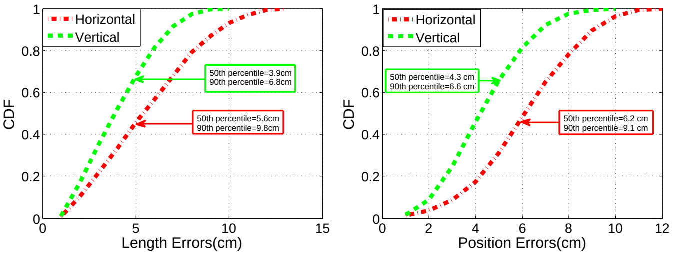
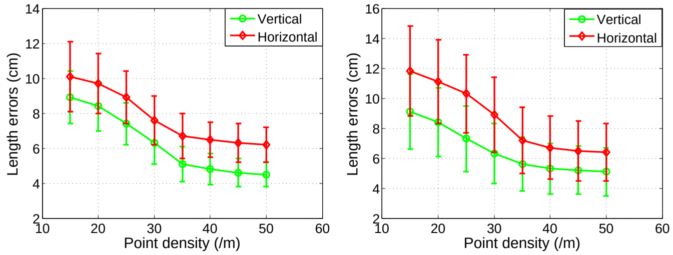


Fig. 17. Pictorial description of layout mapping errors



(a) Length errors of layout mapping (b) Positional Errors of layout mapping

Fig. 18. Length errors and positional errors of layout mapping



(a) Length errors VS Points density (b) Position Errors VS Points density

Fig. 19. Layout performance VS Points density

fortunately, sizes of these objects for in-wall applications are not arbitrary, instead, they are among just several optional choices according to relevant industry standards, especially for pipes and rebars. For the exploration of distinguishing pipes and rebars of several other diameters, we leave it as our future work.

9.3 Issues about layout mapping

For layout mapping, in order to figure out the layout of a in-wall pipeline system, enough points where pipes traverse should be located on the wall. In our layout reconstruction method, we hold a assumption that the length of shortest pipe segment in the pipeline system should not be less than a certain threshold (20 cm in our context). This assumption is practically reasonable but theoretically imperfect, since it is not always a easy job to know about this. As a result, another future job is to try some methods in computational geometry field. Moreover, reference points segments a whole scanning path into multiple parts and reduces scanning distance in each round. This greatly improves user comfort and working efficiency in the stage of collecting sample points for layout mapping. The selection of reference points nearly doesn't affect the performance of SmartScanner according to our tests, while it affects user comfort and sampling efficiency. Generally, it is better to select spatially-separated reference points as the way adopted in our experiments.

10 Conclusions

In this paper, we propose a novel smartphone-based system, SmartScanner, which can distinguish certain objects and map pipeline layouts existing in walls. For distinguishing in-wall objects, experimental results show that SmartScanner runnning on iPhone 4 and Xiaomi Mi2S platforms can achieve no less than an avergae accuracy of 97.6%, 92.8%, 84.2%, when objects are hidden in a wall at a depth of 5 cm, 7 cm and 9 cm, respectively. In addition, experiments for layout mapping conducted on a proof-of-concept testbed measuring 1.8 m× 1.0 m demonstrate that 90% length errors are 9.8 cm and 6.8 cm for horizontal and vertical pipe segments. And 90% position errors of turning points in horizontal and vertical directions are 9.1 cm and 6.6 cm, respectively. By combining multiple in-built sensors, namely, the accelerometer, the gyroscope and the magnetometer, SmartScanner avoids utilizing specialized devices/tools, but can achieve practical functions in a cost-effective way as well.

References

[1] A. Massey, *Interior of the twentieth century*. New York: Thames & Hudson, 1990.

[2] M. M, G. H, and M. M. A, *Pipeline design and construction: a practical approach*, 3rd ed. New York: ASME press, 2007.

[3] Daniels and D. J, *Ground penetrating radar*, 2nd ed. Herts,United Kingdom: The Institution of Engineering and Technology (The IET), Jan 2004, vol. 1.

[4] Engineering and P. S. R. C. (EPSRC), "Mapping the underworld (mtu)," Nov 2008. [Online]. Available: http://www.mappingtheunderworld.ac.uk

[5] B. P. Tools, "Wall scanner gms120," Nov 1886. [Online]. Available: http://www.boschtools.com/Products/Tools/Pages/BoschProductDetail.aspx?pid=GMS120

[6] DEWALT, "Radar scanner kit," Mar 2002. [Online]. Available: http://www.dewalt.com/tools/cordless-instruments-radar-scanning-dct418s1.aspx

[7] T.-t. T. Lai, Y.-h. T. Chen, P. Huang, and H.-h. Chu, "Pipeprobe: a mobile sensor droplet for mapping hidden pipeline," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 113–126.

[8] T. J. N. H., "Pipeline pigging technology," *Gulf Professional Publishing*, 1992.

[9] E. Čermáková, "Magnetization of steel building materials and structures in the natural geomagnetic field," *Acta Polytechnica*, vol. 45, no. 6, 2005.

[10] A. J and Z. B. G., "Magnetic anomaly detection in ferromagnetic material," in *PIERS Proceedings*, 2011.

[11] L. Barrows and J. E. Rocchio, "Magnetic surveying for buried metallic objects," *Groundwater Monitoring & Remediation*, vol. 10, no. 3, pp. 204–211, 1990.

[12] J. E. Mcfee, Y. Das, and R. Ellingson, "Locating and identifying compact ferrous objects," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, no. 2, pp. 182–193, 1990.

[13] J. McFee, M. Bell, B. Dempsey, R. Chesney, and Y. Das, "A magnetostatic signature measurement and analysis system," *Journal of Physics E: Scientific Instruments*, vol. 18, no. 1, p. 54, 1985. [Online]. Available: http://iopscience.iop.org/0022-3735/18/1/015

[14] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *International Conference on Indoor Positioning and Indoor Navigation*, vol. 13, 2012, p. 15th.

[15] N. H. Q. Phuong, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro, "A dcm based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 859–876, 2009.

[16] E. Edwan, J. Zhang, J. Zhou, and O. Loffeld, "Reduced dcm based attitude estimation using low-cost imu and magnetometer triad," in *Positioning Navigation and Communication (WPNC), 2011 8th Workshop on*. IEEE, 2011, pp. 1–6.

[17] S. Ayub, A. Bahraminisaab, and B. Honary, "A sensor fusion method for smart phone orientation estimation," in *Proceedings of the 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool*, 2012.

[18] D. Gebre-Egziabher, R. C. Hayward, and J. D. Powell, "Design of multi-sensor attitude determination systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 40, no. 2, pp. 627–649, 2004.

[19] W. Kang, S. Nam, Y. Han, and S. Lee, "Improved heading estimation for smartphone-based indoor positioning systems," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*. IEEE, 2012, pp. 2449–2453.

[20] Z. Sun, X. Mao, W. Tian, and X. Zhang, "Activity classification and dead reckoning for pedestrian navigation with wearable sensors," *Measurement Science and Technology*, vol. 20, no. 1, p. 015203, 2009.

[21] M. Mladenov and M. Mock, "A step counter service for java-enabled devices using a built-in accelerometer," in *Proceedings of the 1st international workshop on context-aware middleware and services: affiliated with the 4th international conference on communication system software and middleware (COMSWARE 2009)*. ACM, 2009, pp. 1–5.

[22] S. Shin, C. Park, J. Kim, H. Hong, and J. Lee, "Adaptive step length estimation algorithm using low-cost mems inertial sensors," in *Sensors Applications Symposium, 2007. SAS'07. IEEE*. IEEE, 2007, pp. 1–5.

[23] S. Y. Cho, C. G. Park, and H. Y. Yim, "Sensor fusion and error compensation algorithm for pedestrian navigation system," 2003.

[24] J. Kappi, J. Syrjarinne, and J. Saarinen, "Mems-imu based pedestrian navigator for handheld devices," in *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001)*, 2001, pp. 1369–1373.

[25] C. Sydney and B. Julius, *Geomagnetism*. Oxford: Clarendon Press, 1940.

[26] S. Maus, S. Macmillan, S. McLean, B. Hamilton, A. Thomson, M. Nair, and C. Rollins, "The us/uk world magnetic model for 2010-2015," 2010.

[27] J. Lenz and A. S. Edelstein, "Magnetic sensors and their applications," *Sensors Journal, IEEE*, vol. 6, no. 3, pp. 631–649, 2006.

[28] J. E. Mcfee, R. Ellingson, and Y. Das, "A total-field magnetometer system for location and identification of compact ferrous objects," *Instrumentation and Measurement, IEEE Transactions on Instrumentation and Measurement*, vol. 43, no. 4, pp. 613–619, 1994.

[29] J. E. McFee, R. Ellingson, J. Elliott, and Y. Das, "Estimation of depth, orientation, length and diameter of long, horizontal ferrous rods using a fluxgate magnetometer," DTIC Document, Tech. Rep., 1993.

[30] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 3, pp. 319–342, 2002.

[31] Z. Guo, D. Liu, Z. Chen, and H. Meng, "Modeling on ground magnetic anomaly detection of underground ferromagnetic metal pipeline," *Bridges*, vol. 10, pp. 9 780 784 412 619–103, 2014.

[32] N. Amenta, M. Bern, and D. Eppstein, "The crust and the β-skeleton: Combinatorial curve reconstruction," *Graphical models and image processing*, vol. 60, no. 2, pp. 125–135, 1998.

[33] T. K. Dey, K. Mehlhorn, and E. A. Ramos, "Curve reconstruction: Connecting dots with good reason," in *Proceedings of the fifteenth annual symposium on Computational geometry*. ACM, 1999, pp. 197–206.

[34] L. H. De Figueiredo and J. de Miranda Gomes, "Computational morphology of curves," *The Visual Computer*, vol. 11, no. 2, pp. 105–112, 1994.

[35] F. Bernardini and C. L. Bajaj, "Sampling and reconstructing manifolds using alpha-shapes," 1997.

[36] D. Attali, "r-regular shape reconstruction from unorganized points," in *Proceedings of the thirteenth annual symposium on Computational geometry*. ACM, 1997, pp. 248–253.

[37] T. K. Dey and P. Kumar, "A simple provable algorithm for curve reconstruction," in *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1999, pp. 893–894.

[38] T. K. Dey, K. Mehlhorn, and E. A. Ramos, "Curve reconstruction: Connecting dots with good reason," in *Proceedings of the fifteenth annual symposium on Computational geometry*. ACM, 1999, pp. 197–206.

[39] Y.-H. Chu, J.-W. Huang, K.-T. Chuang, D.-N. Yang, and M.-S. Chen, "Density conscious subspace clustering for high-dimensional data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 1, pp. 16–30, 2010.

PLACE
PHOTO
HERE

Yongpan Zou received the BEng degree of Chemical Machinery from Xi'an Jiaotong University, Xi'an, China. Since 2013, he is a PhD student in the department of computer science and engineering in Hong Kong University of Science and Technology (HKUST). His current research interests mainly include: wearable computing, mobile computing, and wireless communication.

PLACE
PHOTO
HERE

Guanhua Wang received his B.Eng. degree from Southeast University, Nanjing, China, in 2012, and obtain a master of philosophy degree in computer science and engineering at Hong Kong University of Science and Technology. His main research interests in HKUST include Wi-Fi radar, wireless communications and mobile computing. He is now pursuing his Ph.D. degree at AMPLab in Computer Science Division at UC Berkeley since Fall 2015.

PLACE PHOTO HERE

Kaishun Wu received the BEng degree in Microelectronics from Sun Yat-sen University, Guangzhou, China and the PhD degree in Computer Science and Engineering from Hong Kong University of Science and Technology, in 2007 and 2011, respectively. After that, he worked as a research assistant professor in the Hong Kong University of Science and Technology. Dr. Wus research interests include wireless communications and mobile computing. He has co-authored 2 books and published over 60 refereed papers in international leading journals and primer conferences. He is the inventor of 5 US and 35 Chinese pending patents (9 are issued). He won the best paper awards in IEEE Globecom 2012, IEEE ICPADS 2012 and IEEE MASS 2014. He received 2014 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award and selected as 1000 Talent Plan for Young Researchers.

PLACE PHOTO HERE

Lionel M. Ni is Chair Professor in the Department of Computer and Information Science and Vice Rector of Academic Affairs at the University of Macau. Previously, he was Chair Professor of Computer Science and Engineering at the Hong Kong University of Science and Technology. He received the Ph.D. degree in electrical and computer engineering from Purdue University in 1980. As a fellow of IEEE and Hong Kong Academy of Engineering Science, Dr. Ni has chaired over 30 professional conferences and has received eight awards for authoring outstanding papers.