PipeProbe: A Mobile Sensor Droplet for Mapping Hidden Pipeline

Tsung-te (Ted) Lai[1] **Yu-han (Tiffany) Chen**[1] **Polly Huang**[2, 3] **Hao-hua Chu**[1,2]

Department of Computer Science and Information Engineering[1], Graduate Institute of Networking and Multimedia[2], Department of Electrical Engineering[3]

National Taiwan University, Taipei, Taiwan

{f96922152, yuhan, hchu}@csie.ntu.edu.tw, phuang@cc.ee.ntu.edu.tw

Abstract

This paper presents PipeProbe, a mobile sensor system for determining the spatial topology of hidden water pipelines behind walls. PipeProbe works by dropping a tiny wireless sensor capsule into the source of the water pipelines. As the PipeProbe capsule traverses the pipelines, it gathers and transmits pressure and angular velocity readings. Through spatio-temporal analysis on the sensor readings, our algorithm locates all turning points in the pipelines and maps their 3D spatial topology. We evaluated the PipeProbe system by developing a prototype and using data collected in our experimental testbed. Results show that the PipeProbe system successfully located and estimated 90% of all pipe tube lengths within 8-cm accuracy on average tube lengths of 76 cm. PipeProbe also successfully located 90% of all turning points within 15-cm accuracy on average length paths of 335cm.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and Embedded Systems, Signal Processing Systems.

General Terms

Design, Experimentation, Performance.

Keywords

Wireless Sensor Networks, Mapping Water Pipeline, Sensor Inference, Constraint Satisfaction.

1. Introduction

Houses are often equipped with an extensive water pipeline network distributing water to different water-using fixtures and appliances throughout the home, such as bathroom toilets, kitchen faucets, garden sprinklers, washing machines, etc. It is therefore unfortunate that plumbing is ranked as one of the ten most frequently found problems in homes [11]. Leaking pipes are one of the most common problems in plumbing [12], and hidden leaking pipes often cause extensive damage to floors, walls and belongings in a home.

The first step in fixing leaking pipes is to locate where they are for further inspection. When leaking water pipes are hidden inside walls and underneath floors, diagnosing their location without direct inspection becomes very difficult, especially when the original diagram of the pipeline layout is also missing. Searching for the pipeline locations becomes guesswork and often requires a brute-force method, such as knocking down walls and stripping floor coverings. This problem created an opportunity for the development of PipeProbe, a mobile sensing probe that is dropped into the source of the water pipeline. During its traversal of pipeline, the PipeProbe collects the sensor readings (i.e., pressure and angular velocity) necessary for the reconstruction of the 3D spatial topology of the traversed water pipeline. In comparison to the traditional brute-force approach, the PipeProbe system is a non-intrusive method of mapping and locating indoor water pipelines that requires no alteration to the water pipeline infrastructure. Since leaking often occurs at places where two disjoint pipe tubes join together, mapping locations of these pipeline turning points is especially important for inspection.

Three previous projects that applied wireless sensor network technologies for monitoring water pipes include the NAWMS project [2], the PIPENET project [1] and HydroSense [3]. The NAWMS project detected and located pipe leaks by attaching vibration sensors to the pipe surface. Similarly, the PIPENET project [1] monitored water flow and detected leaks by attaching acoustic and vibration sensors to large bulk-water pipelines and pressure sensors to normal pipelines. HydroSense [3] employed a single endpoint sensing solution in which the amount of water outflow from each water outlet could be uniquely estimated by learning and recognizing a pressure wave signature. In contrast to these projects, the PipeProbe system adopts a mobile sensing approach. It employs a tiny mobile sensor that travels inside of the water pipeline infrastructure while remotely performing on-the-spot data collection near possible problematic locations. Alvarado et al. [9] developed a robotic fish under a foot long that closely mimics a real fish's natural swimming motion. This robofish is equipped with sensors to detect environmental pollutants. Its one-foot size is considerably larger than that of our PipeProbe capsule, and its motor requires a 2.5-5W external power source.

SenSys10, 03-NOV-2010, Zurich, Switzerland

The three important contributions of this work are the following:

- Rather than fixing sensing points in the utility infrastructure, PipeProbe adopts a mobile sensing approach in which a mobile sensor travels and performs on-the-spot data collection at different places.
- A novel localization method was developed to accurately estimate the 3D spatial topology of the capsule-traversed water pipelines from the pressure and rotation graphs collected and computed by the PipeProbe system. Experimental results from our testbed showed that our mobile sensing approach produced a high-precision 3D map of the pipeline with centimeter-level errors.
- Since the PipeProbe capsule is designed to model a water droplet, its physical movement leverages the force inside of the pipeline infrastructure for propulsion. This means that no motoring is necessary to power its movement, which increases the PipeProbe capsule's energy-efficiency and allows it to operate on only 15 mA of current. To illustrate, a tiny lithium button cell battery can keep our PipeProbe capsule operating for over 1 kilometer at a water flow rate of 15 centimeters per second.

The rest of this paper is organized as follows. Section 2 presents the design principles for PipeProbe's mobile sensing approach. Section 3 explains the design and implementation of PipeProbe's sensing capsule and the process of data collection. Section 4 details the PipeProbe's system of operation and how data processing is used to map the spatial topology of the pipelines. Section 5 describes the experimental testbed and scenarios. Section 6 presents the evaluation's results. Section 7 discussion limitations and their possible solution. Section 8 reviews related work. Finally, Section 9 concludes the study and suggests directions for future studies.

2. Pipeline Profiling

Ideally, the PipeProbe system would be to clone a micro sensing hydro molecule that flows along the pipeline, like the myriad other hydro molecules in the fluid system, and observe the wall-embedded pipelines from within. The current PipeProbe prototype is made of a tiny wireless sensor node packaged in a water-proof spherical shell measuring 4 centimeters in diameter. PipeProbe works in two stages: (1) *data collection stage*, and (2) *data analysis stage.* In the data collection stage, the PipeProbe capsule traverses a water pipeline and collects data from the pressure and gyroscope sensors; in the data analysis stage, our system analyzes the sensor readings and derives the 3D spatial topology of the traversed water pipeline.

PipeProbe operates as follows. First, the capsule is dropped into the main water inlet of a home or building. When an outlet (i.e., a faucet) is opened, the force of the resulting water flow pushes the capsule through different possible paths for the connected water pipes. While the capsule is flowing inside the water pipelines, it logs the sensed pressure and angular velocity data to an EEPROM. A radio within the PipeProbe capsule transmits the sensor data buffered in the EEPROM to a PC-connected base station. Alternatively, when the PipeProbe capsule flows out of a water outlet, users can manually transfer sensor data from the capsule's EEPROM to a PC. Finally, the data analysis part of the PipeProbe system computes and maps the 3D spatial topology of the hidden water pipeline.

During the data collection stage, if the PipeProbe capsule flows out of a water outlet, it can be reinserted into the water inlet and reused for additional data collection. Multiple trips enable the discovery of diverse pipeline branches, which are used for producing the full map. In addition, multiple measurements over the same flow path can be utilized to filter out noise in the data and enhance the accuracy of the 3D spatial topology reconstruction.

2.1 Vertical Movement

The water pressure sensor is based on the Pressure Principle, which states that static pressure at any sensing point in a confined liquid is produced by the weight of the liquid above that point. In other words, this pressure depends only on the height of the liquid above that sensing point and the liquid density. If a liquid is confined in a tank, the pressure at any sensing point in the tank is given by:

$$P = \rho g h \qquad (1)$$

where P is the pressure, ρ is the density of the liquid (in our case, water), g is the acceleration due to gravity and h is the height of the sensing point. With constant gravity and density, pressure is proportional to the height of the sensing point.

Based on the Pressure Principle, the movement on the vertical plane of a pressure-sensing capsule can be estimated from the pressure difference between two sensing points. Consider a vertical pipe with length L, the difference in pressure readings between the top and bottom of the vertical pipe is ΔP. From equation (1), the pressure difference is

$$\Delta P = \rho g \Delta h \qquad (2)$$

Since Δh is the length of the vertical pipe, L can be derived as follows

$$L = \Delta P / \rho g \qquad (3)$$

2.2 Horizontal Movement

Since a capsule detects no pressure difference while traveling on a horizontal plane, the Pressure Principle is only applicable to determining the capsule's traversal time on a vertical plane. Another approach based on angular velocity from a gyroscope sensor is used to estimate the capsule's movement direction on a horizontal plane. Combining both length and direction of horizontal movement gives the full 2D horizontal mapping. We will first describe how to

determine a pipe's horizontal length from the capsule's traversal time on a horizontal plane, and then how to locate the horizontal turning points from the capsule's angular or rotation velocity.

To determine the length of a horizontal tube (h-tube), the sensor is cased in a spherical shell and its density (i.e., weight over volume) is adjusted so that it equals the water's density. This allows the capsule to flow through the pipes as if it were part of the fluid system. As a result, the estimated water flow velocity approximates the capsule's own velocity. This allows us to measure the duration that the pressure sensor's readings remain constant, giving us the length of the corresponding horizontal pipe (L) which is estimated by multiplying the capsule's flow velocity (v) by the flow time (t).

$$L = v * t \qquad (4)$$

For this calculation to work, we make the assumption that the diameter of the pipes is uniform; thus, water flow velocity in the horizontal plane is constant across all connecting pipes. To derive the water flow velocity, one can fix the valve at the water's inlet and then divide the amount of water entering the inlet and the area size of the pipe's intake surface. Since home water pipes come in several selected sizes [10], our future work will discuss how to relax this assumption using additional sensors on the PipeProbe capsule.

To locate a pipe's horizontal turning points, a gyroscope on the PipeProbe capsule measures its angular velocity. By integrating angular velocity into the rotation angle, the PipeProbe system distinguishes when the capsule makes a left horizontal turn, i.e., with the positive 90-degree rotation angle, from a left horizontal turn, i.e., with a negative 90-degree rotation angle.

3. Data Collection

The PipeProbe capsule was prototyped with the Eco wireless sensor mote [4]. The Eco mote is an ultra-compact and low power wireless sensor node. It measures only 13 mm (L) × 11 mm (W) × 7 mm (H) and weighs 3 grams (including battery). It consumes less than 10 mA in transmission mode (0 dBm) and 22 mA in receiving mode. Its maximum data rate and RF range are 1Mbps and 10 meters, respectively. The Eco's small form factor and low power consumption make it ideal for our PipeProbe capsule which requires a tiny size to allow it to flow freely inside a water pipeline. The Eco mote has a flexible-PCB type expansion port that has 16 pins. This expansion port includes two digital I/O pins, two analog input lines, serial peripheral interface (SPI), RS232, and voltage inputs for a regulator and battery charging. The Intersema MS5541C pressure sensor [13] is wired to the Eco mote via the SPI protocol. MS5541C measures a pressure range from 0 to 14 bars with a resolution of 1.2 mbars. Given less than 5 uA operating current the MS5541C enables the Eco mote to sample frequently without drawing too much battery power. The MS5541C requires an oscillator at the frequency of 32.768 kHz for sensor ADC. To fulfill that requirement a SG3030JC was chosen for the external oscillator. The pressure sensor samples the water pressure at a peak rate of 33 Hz. Figure 1 shows the components in the PipeProbe capsule.

After the pressure sensor and oscillator were integrated with the Eco mote, they were enclosed with a waterproof plastic casing. The pressure sensor is exposed outside of the casing to maintain contact with the water. This packaging went through 4 iterations of design. The first prototype (Figure 2) used a cylindrical casing. However, the cylinder shape (which has non-uniform surfaces from different perspectives) proved problematic, incurring varying moving velocities as the capsule tumbled through the pipes. In the 2nd iteration we changed the case to a spherical shape to solve this problem.

The next problem that we discovered was the weight of the capsule. The electronics and the case were too light. The density difference between the capsule and water resulted in a constantly floating capsule, whose traveling velocity was particularly unstable. The 2nd prototype failed to behave like a water droplet, i.e., travels at the same velocity as the current. Thus, a counterweight was added to the third prototype so that while the capsule is sitting still in the water it will neither float to the surface nor sink to the bottom. Given the target density at 1g/cm^3, the ideal weight was 33.51 grams for a 2-cm radius sphere.

However, the 3rd prototype still required some modifications, as indicated by significant variations in the pressure readings. This was due to the fact that the pressure sensor may turn arbitrarily as the capsule rotated through the pipelines. To minimize the jitter in the pressure sensor readings, the 4th prototype (Figure 3) fixed the counterweight to the bottom hemisphere of the capsule. This design minimized the amount of flipping rotation on the capsule around the z-axis. The idea is like a roly-poly toy, or a tumbler, which has a heavier hemisphere below its center. When the tumbler is pushed down, it quickly rights itself. Simple tests confirmed that creating a heavier hemisphere in the capsule significantly reduced the amount of flipping rotation and stabilized the pressure sensor's readings.

Figure 4 shows our 5th and final prototype, which incorporated a gyroscope module (Figure 1(f)) for detecting horizontal turns, thus giving it 3D pipeline mapping capability. The gyroscope module is the STMicroelectronics LISY300AL chip [6], which measures the rotational motion along the yaw (z) axis with a ±300°/s range and outputs an analog voltage. The gyroscope module is fixed precisely at the top of one of the capsule's hemispheres such that the gyroscope lays flat on the horizontal plane in order to obtain an accurate z-axis measurement. Furthermore, the final prototype has a tail-like fin whose function is to further stabilize the capsule's movement on the horizontal plane and to re-align the capsule's heading in the presence of turbulent water flow within the pipeline.

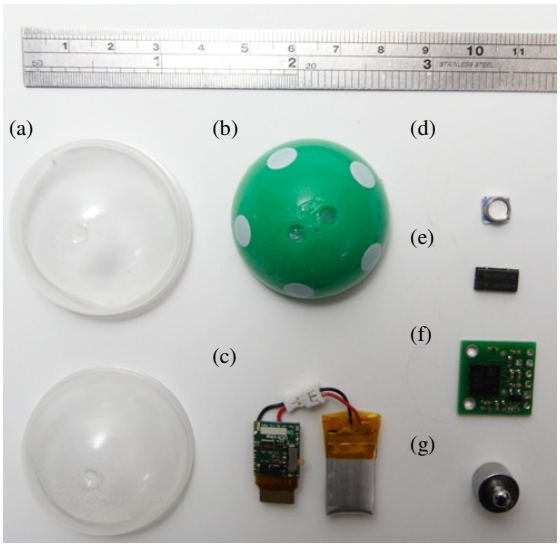

Figure 1. The PipeProbe capsule and its parts: (a) waterproof plastic casing, (b) hemisphere used to stabilize flow velocity, (c) Eco mote, (d) a pressure sensor, (e) an oscillator, (f) a gyroscope sensor, and (g) counter weight.

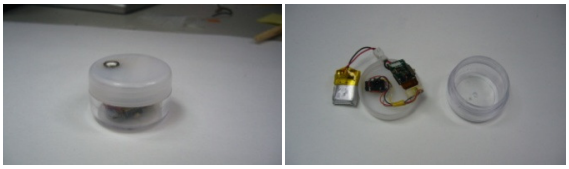

Figure 2. The initial capsule prototype.

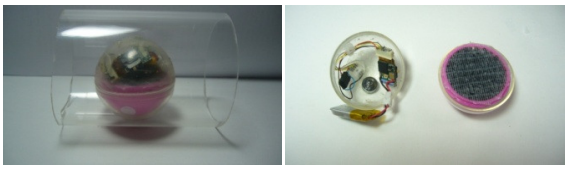

Figure 3. The 4th capsule prototype.

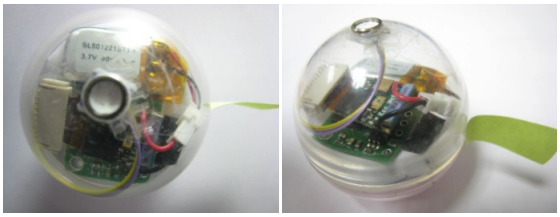

Figure 4. The final capsule prototype.

4. Data Processing

Figure 5 shows four steps in the analysis of the collected data from PipeProbe capsule. (1) A *median filter* is applied to smooth out and remove noises from the pressure time-series data. (2) *Turn detection* performs a spatial-temporal analysis on the pressure and gyroscope time-series data to detect all vertical and horizontal turning points on the flow path of the PipeProbe capsule. (3) Since the sensor data alone cannot determine the precise location of all turns, *Layout mapping* solves for unknown coordinates of these turns by modeling it as a constraint satisfaction problem in which the constraint specifies that the coordinates of these intermediate turns must fall on a path between the known coordinates of the inlet and outlet. Additionally, repeated measurements from multiple mapping trips are aggregated to remove noisy outliers and enable more accurate reconstruction of the 3D spatial topology of pipelines. (4) Solving the constraint satisfaction problem may generate multiple topological solutions. To find the correct spatial topology, the PipeProbe system first uses the spatial constraints within a home's walls to eliminate unreachable placements whose topologies do not fit within the confined spaces of the walls. Furthermore, beacon listeners are placed on walls that have the remaining ambiguity paths if needed. The listener on the wall which has the correct path where the PipeProbe capsule actually flows by would measure the highest received packet rate. These four steps are elaborated below.

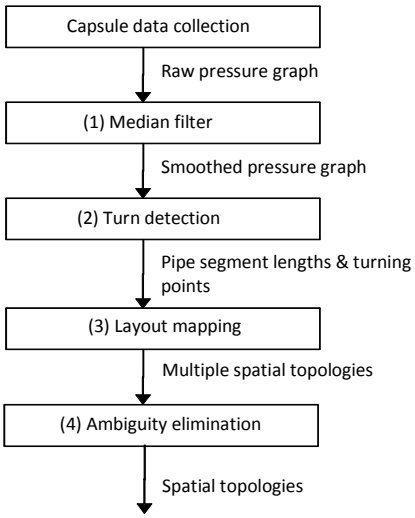

Figure 5. Data Analysis

4.1 Median Filter on Pressure Reading

Median filtering is a common technique for removing noises in image processing, and is applied here to smooth the pressure signal. We first divide the pressure signal into windows of ten pressure samples. The median of the pressure values is computed within each window. The medians form the skeleton of the smoothed signal. The true pressure signal is very likely segment-wise linear. Thus, we reconstruct the intermediate data points of the smoothed signal by linear interpolation of the consecutive medians. To illustrate, Figure 6 shows a raw pressure signal where *x*-axis represents the time the pressure sensor is sampled and the *y*-axis depicts the pressure reading at the time. Applying the median filter produces a smooth pressure signal in Figure 7.

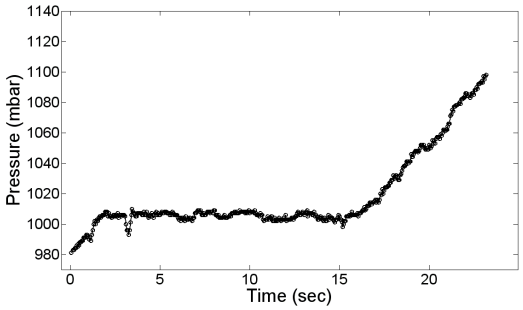

Figure 6. A raw pressure signal before applying the median filter.

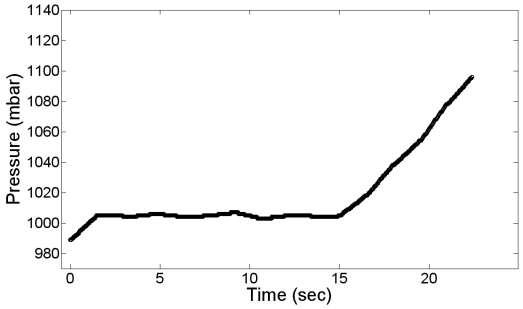

Figure 7. A smoothed pressure signal after applying the median filter.

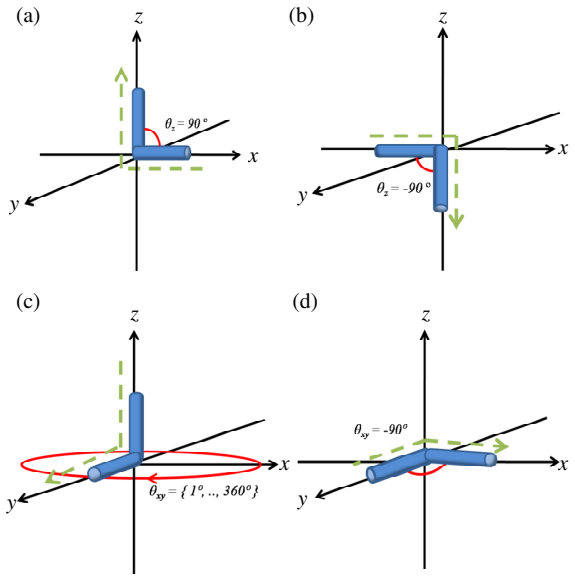

Figure 8. Three types of turns detected by the PipeProbe system. (a)(b) contain two h-v-turns where a pipeline turns from a horizontal plane to a vertical plane and the vertical turning angle is restricted to 90-degree upward or downward. (c) shows a v-h-turn where a pipeline turns from a vertical plane to a horizontal plane and the horizontal turning angle has unrestricted freedom from 1 to 360-degrees. (d) gives an h-h-turn where a pipeline turns and stays on the same horizontal plane and the horizontal turning angle is restricted to right or left 90-degree.

4.2 Turn Detection

A water distribution pipeline infrastructure consists of multiple rigid tubes and joints (i.e., turning points). Section 2 described the general approach to determine tube length. Here, we define target types of turning points in the PipeProbe system and describe the corresponding turn detection algorithms.

The PipeProbe system detects three types of turning points in the pipeline infrastructure:

- *h-v-turn(t, p, θ_z)* or horizontal-to-vertical turn: Figure 8(a)(b) contain two examples of horizontal tubes making an θ_z vertical turn either upward or downward along the *z*-axis. At this vertical turning point, PipeProbe measures the pressure reading *p* at time *t*. θ_z is restricted to be either a negative 90-degree depression angle or a positive 90-degree elevation angle, i.e., θ_z ={-90º, 90º}. From consultation with a master plumber, this 90-degree vertical turning restriction follows the conventional residential pipeline layout guide. This convention is also consistent with the example piping layouts recommended by PPFA [10] for four most common house types.
- *v-h-turn(t, p, θ_{xy})* or vertical-to-horizontal turn: Figure 8(c) shows an example of a vertical tube making an θ_{xy} horizontal turn. At this horizontal turn, PipeProbe measures the pressure reading of *p* at time *t*. θ_{xy} has an unrestricted 360-degree freedom on the horizontal plane, i.e., θ_{xy} ={1º, .., 360º}.
- *h-h-turn(t, θ_{xy})* or horizontal-to-horizontal turn: Figure 8(d) shows an example of a horizontal tube making an θ_{xy} horizontal turn at time *t*. θ_{xy} is restricted to be either a positive 90-degree left angle or a negative 90-degree right angle, i.e., θ_z ={90º, -90º}. Since 90-degree pipe joints are the most commonly found (or only available) joints in water pipeline supply stores, this work focuses on mapping pipelines that make 90-degree turns.

We developed v-turn and h-turn detection algorithms to identify and locate the above three turn types. The v-turn detection algorithm locates (1) v-h-turns and (2) h-v-turns by analyzing the change in pressure readings. The h-turn detection algorithm identifies (3) h-h-turns by processing and integrating angular velocities from a gyroscope to give the pipe's horizontal rotation. The following subsections describe the details of these two turn detection algorithms.

4.2.1 V-Turn Detection

The v-turn detection algorithm locates v-h-turns and h-v-turns from the smoothed pressure signals obtained in the previous step. At the same time, it also computes lengths of the pipe tubes and directions of turns. When the capsule is moving vertically, the pressure increases or decreases linearly over the distance traveled. In contrast, when the capsule is moving horizontally, the pressure level stays constant

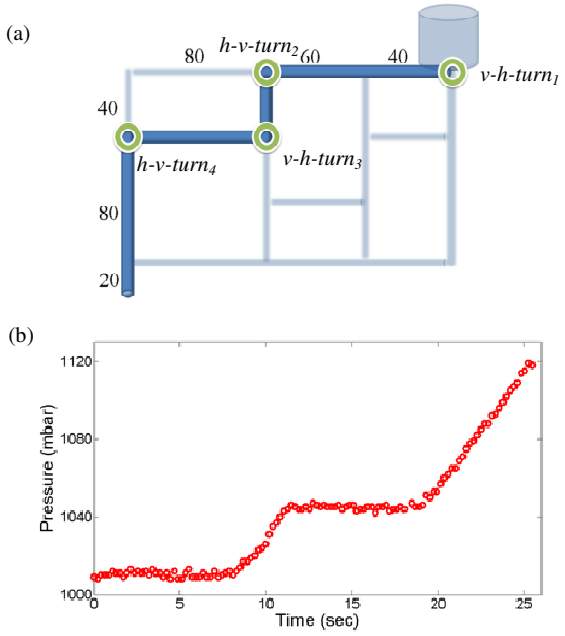

Figure 9. This example demonstrates how the v-turn detection algorithm works. (a) shows an example pipeline structure (only the red part) containing two h-v-turns and one v-h turn. (b) gives the corresponding pressure graph collected by the PipeProbe capsule.

regardless of the distance traveled. In fact, (1) each turning point on the pressure graph marks a v-turn in the physical pipeline topology. (2) Distance traveled between two adjacent v-turns is the length of a vertical tube (v-tube). These two mapping rules are explained in detail as follows.

H-v-turns, v-h-turns and their connected tubes are often hidden behind a vertical 2D wall space. Changes in the pressure signal occur while the PipeProbe capsule is traversing an h-v-turn or a v-h-turn. Figure 9(a) shows a pipeline structure, i.e., the red flow path, containing a downward h-v-turn, followed by a v-h-turn and another downward h-v-turn. Figure 9(b) illustrates the collected pressure graph during the capsule's traversal of this pipeline structure. The pressure graph shows a steady rise of pressure readings after the capsule completes its first downward h-v-turn, i.e., turning from horizontal movement (constant pressure) to downward movement (increasing pressure). The rise in pressure readings comes to a halt after the capsule makes a v-h-turn, i.e., turning from downward movement (increasing pressure) to horizontal movement (constant pressure). Finally, the pipeline structure makes a downward h-v-turn, i.e., turning from horizontal movement (constant pressure) to downward movement (increasing pressure). By recognizing different pressure changing shapes in the pressure graph, the v-turn detection algorithm locates not only these turning points but also the upward/downward direction of v-h-turns and h-v-turns.

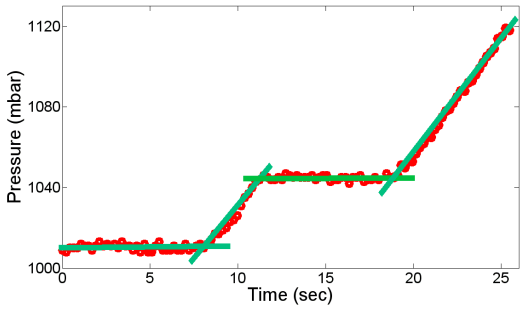

Figure 10. The green line segments depict the resulting linear fit from the best combination of the candidate turning points.

The v-turn detection algorithm takes the following two steps. (1) Compute the derivatives from the smoothed pressure graph using a sliding window. The horizontal path corresponds to the derivative of the smoothed data equal to zero, and the derivative of the vertical path may be either larger or smaller than zero depending on the direction of the vertical path. (2) After segmenting the horizontal and vertical paths and identifying the direction of vertical movement, we then try to pinpoint the exact turning point at the intersection of the zero and non-zero derivative segments.

There are a number of candidate points for the intersections. A simple solution is to identify the data point where the change of the derivatives is the highest. This, however, is sensitive to pressure sensing noise. To minimize estimation error, we define a set of candidate points for each intersection. When the derivative shows a vertical-horizontal movement, we include the data point that gives the largest derivative change and the four preceding data points in the candidate set. When the derivative shows a horizontal-vertical movement, we include the largest change point, as well as the four subsequent data points. Experiments revealed that a window size of 5 samples provided the best result.

There are K candidate sets for K intersections. Each intersection is an h-v-turn or a v-h-turn in the physical pipelines. Having 5 data points in each candidate set, there are 5^K combinations to search for the best solution. For each combination, we derive the best linear fit by regression for each of the segments. By testing all combinations and computing the mean square error of the individual data points for the best linear fit, we can identify the combination such that the sum of the mean square error over all segments is the minimum. Figure 10 shows the turn detection results from Figure 9(b).

Here we show an example of the v-turn detection algorithm detecting the blue-colored pipe segment in Figure 9. First, the turning point notation in Section 4.2 is used to specify sensor data collected on the two turning points: *h-v-turn*$_2$ and *v-h-turn*$_3$. For example, *h-v-turn*$_2$*(8.27 sec, 1,012 mbar, -90º)* means that the PipeProbe capsule sensed a pressure reading of 1,012 bar at the time point 8.27

seconds with an inferred turning angle of *-90º* from the changes in the pressure signals.

h-v-turn$_2$*(8.27 sec, 1,012 mbar, -90º)*
→ *v-h-turn*$_3$*(11.46 sec, 1,046 mbar,* θ_{xy3}*)*

Applying equation (3) gives the length of the blue v-tube between *h-v-turn*$_2$ and *v-h-turn*$_3$. That is, the 34 mbar pressure difference (1,046 mbar – 1,012 mbar) between the two turning points approximates to 40.34 cm drop in vertical height. Therefore, the corresponding v-tube is denoted as follows:

h-v-turn$_2$*(-90º)* → *v-tube*$_2$*(40.34 cm)* → *v-h-turn*$_3$(θ_{xy3})

4.2.2 H-Turn Detection

The h-turn detection algorithm locates h-h-turns and their horizontal turning angles (θ_{xy}) from pressure and gyroscope sensor readings. H-h-turns and their connected tubes are often hidden under floors and above dropped ceilings. Since 90-degree pipe joints are the most commonly found (or the only available) joints in water pipeline supply stores, the h-turn detection algorithm focuses on detecting 90-degree right and left h-h-turns, i.e., θ_{xy} = {90º, -90º}. The h-turn algorithm takes the following three steps: (1) identifying horizontal tubes, (2) applying a threshold-based filter to remove noises in the angular velocity readings from the gyroscope, and (3) calculating the capsule's rotation rate and identifying the h-h-turn. Figure 11(a) shows an example pipeline structure, i.e., the red flow path, containing a v-h-turn followed by two h-h-turns. This example illustrates how the h-turn detection algorithm detects these h-h-turns.

In the first step, h-tubes are recognized from the pressure readings (Figure 11(b)) as described in Section 4.2.1. When the PipeProbe capsule is traveling on the horizontal plane, the capsule detects little or no pressure difference as the height of the capsule stays unchanged.

The second step filters out noises in the raw angular velocity data and keeps only those angular velocity readings where the capsule passes through an h-h-turn, so that we can produce the correct rotation rate. The filtering of the raw angular velocity graph (Figure 11(c)) occurs in two stages. First, any high angular velocity readings during PipeProbe's traversal of v-tubes are considered noises because h-turns by definition do not occur in v-tubes. Second, a simple threshold-based filter is applied to remove small random noises from the raw angular velocity readings. When the capsule flows inside h-tubes, its gyroscope sensor may measure relatively small angular velocity due to water turbulence inside the tube. Experiments showed that a threshold value of ±100 deg/sec is effective in filtering out angular velocity noises from the gyroscope. Therefore, if the angular velocity is within ±100 deg/sec, we can simply ignore it. Figure 11(d) shows the resulting filtered angular velocity graph.

The third step calculates the rotation rate (Figure 11(e)) by integrating the filtered angular velocity graph (Figure 11(d)). Interestingly, the experiment results show that when a PipeProbe capsule makes a 90-degree left h-h-turn, its rotation angle reveals this unique angular velocity pattern – first exhibiting a high positive signal (i.e., a positive value

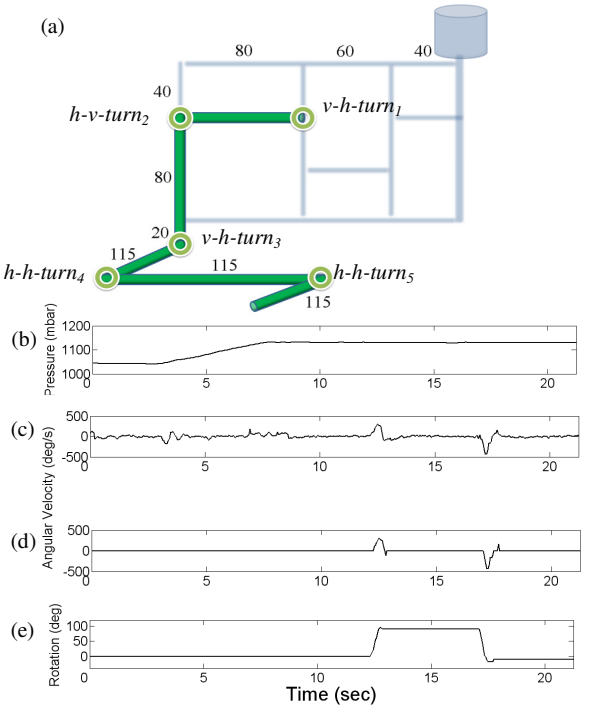

Figure 11. This shows how the h-turn detection algorithm works. (a) is an example pipeline structure (only the red part) containing a v-h-turn followed by two h-h-turns, (b) is the corresponding pressure graph collected by the PipeProbe capsule, (c) is the raw angular velocity graph produced by the gyroscope sensor on the PipeProbe capsule, (d) is the noise-filtered angular velocity graph from (c), and (e) is the rotation angle of capsule graph calculated from (d). The water flow velocity was at 24 cm/sec.

corresponds to leftward angular velocity) and followed by a low negative signal (i.e., a negative value corresponds to rightward angular velocity). This "high-positive-low-negative" angular velocity pattern matches the actual observation on how the PipeProbe capsule makes a right h-h turn due to our tail-like fin design in our final prototype. First, the water flow at the turning joint pushes the capsule to over-rotate to the right. Then, the capsule corrects its heading direction by making a moderate rotation in the reverse-left direction. On the other hand, when the capsule makes a 90-degree right h-h-turn, it procures a "high-negative-low-positive" angular velocity pattern. Last but not least, for h-h-turn detection, we select the peak positive/negative signal from the filtered angular velocity data for the left/right turn on horizontal plane.

Here we show an example of the h-turn detection algorithm detecting the green-colored pipe segment in Figure 11. First, the turning point notation in Section 4.2 is used to specify sensor data collected on the two turning points: *h-h-turn*$_4$ and *h-h-turn*$_5$. For example, *h-h-turn*$_4$*(12.5 sec, 90º)* means that the PipeProbe capsule sensed a turning angle of *90º* from changes in the pressure signals at the time point 12.5 seconds.

$$h\text{-}h\text{-}turn_4(12.5\ sec,\ 90^o) \rightarrow h\text{-}h\text{-}turn_5(17.4\ sec,\ -90^o)$$

After applying Equation (4) with water flow velocity measured at 24 (cm/sec) gives the length of the green h-tube between *h-h-turn*$_4$ and *h-h-turn*$_5$. That is, the 4.9 seconds time difference (17.4 sec – 12.5 sec) between the two turning points multiplying the water flow velocity of 24 cm/sec approximates 117.6 cm of horizontal pipe length.

$$h\text{-}h\text{-}turn_4(90^o) \rightarrow h\text{-}tube_4(117.6\ cm) \rightarrow h\text{-}h\text{-}turn_5(-90^o)$$

4.3 Layout Mapping

The turn detection algorithm in Section 4.2 produces the following turn-tube sequence:

$$\ldots turn_i(\theta_i) \rightarrow tube_i(L_i) \rightarrow turn_{i+1}(\theta_{i+1}) \rightarrow tube_{i+1}(L_{i+1}) \rightarrow \ldots$$

L_i is the length of the tube. For most vertical and horizontal turns, θ_i is a known value (±90º vertical angle) determined by sensing either positive or negative pressure change in the v-turn detection algorithm or by sensing either a positive or negative 90-degree rotation angle in the h-turn detection algorithm. A special turn with an unknown horizontal angle, θ_{xyi}, occurs when an h-tube is preceded by a v-h-turn such as the *v-h-turn*$_3$ in Figure 11.That is, θ_{xyi} can be any value in {1º ~ 360º} and it will be solved by the constraint satisfaction algorithm described as follows.

By analyzing the turn-tube, layout mapping produces a 3D spatial diagram as the result. It works as follows. First, the known position (p_0) of the starting point (i.e., the water inlet) and the known position (p_n) of the end point (i.e., the faucet outlet) are inserted at the beginning/end of this tube -tube sequence.

$$inlet(p_0) \ldots \rightarrow turn_i(\theta_i) \rightarrow tube_i(L_i) \rightarrow \ldots outlet(p_n)$$

Next, layout mapping transforms this turn-tube sequence with inlet/outlet into a constraint satisfaction problem. The model constraint is that the pipeline network must start from $p_0(x_0, y_0, z_0)$, i.e., the position of the inlet into which the PipeProbe capsule is dropped, then move through intermediate vertical/horizontal pipe tubes of various lengths, and finally reach $p_n(x_n, y_n, z_n)$, i.e., the position of the outlet where the PipeProbe capsule flows out.

We can deconstruct a pipeline structure into layers by cutting it from each h-v turn. Hence, each layer begins with a v-h turn followed by one or more h-tubes. We can describe the *x-*, *y-,* and *z*-axis movement on one layer using the following three equations,

x:

$$\sum_{i=0}^{m} \left(L_i^{h-tube}\right) cos(\theta_{xyi}) \quad (5)$$

y :

$$\sum_{i=0}^{m} \left(L_i^{h-tube}\right) sin(\theta_{xyi}) \quad (6)$$

z :

$$\pm \left(L_i^{v-tube}\right) \quad (7)$$

where L_i^{h-tube} denotes the *i*-th horizontal tube in a layer, L_i^{v-tube} denotes the *i*-th vertical tube in a layer, and *m* is the number of h-h turns in a layer.

Summing up all *x*-axis movements from all *n* layers of pipeline structures connects the inlet's starting *x*-position (x_0) to the outlet's ending *x*-position (x_n), thus giving the following *x*-axis constraint satisfaction equation; similar constraint satisfaction equations are derived for *y*- and *z*-axes.

$$x_n = x_o + \sum_{all\ layers} \sum_{i=0}^{m} \left(L_i^{h-tube}\right) cos(\theta_{xyi}) \quad (8)$$

$$y_n = y_o + \sum_{all\ layers} \sum_{i=0}^{m} \left(L_i^{h-tube}\right) sin(\theta_{xyi}) \quad (9)$$

$$z_n = z_o + \sum_{all\ layers} \left[\pm \left(L_i^{v-tube}\right)\right] \quad (10)$$

Another way to understand the above constraint satisfaction equations is as follows. Since h-tubes move pipeline only on the horizontal (*xy*) plane and v-tubes vary only the elevation (*z*-axis) of the pipeline, constraint satisfaction equations can be specified individually on the *x*-, *y*-, and *z*-axes. That is, chaining and summing all positive and negative x-axis movements from all h-tubes must get the pipeline from the starting inlet's *x*-position, i.e., x_0, to the outlet's *x*-position, i.e., x_n, and similarly for *y*-axis and *z*-axis movements.

This constraint satisfaction problem is formally defined here. Let $X_1, X_2,\ldots,$ and X_k be a set of variables, and $C_1, C_2,\ldots,$ and C_m be a set of constraints. Each variable X_i has a non-empty domain D_i of possible values. The goal is to find all possible solutions under all of the constraints. We formulate the variables, values and constraints as follows:

-Variable: $\{\theta_{xyi} \mid i = 1\ldots n\}$, where *n* is the number of turns with unknown turning angles

-Domain: *{1º, 2º,…, 360º}*

-Constraints: Equation (8) and (9)

With the starting point of the pipeline structure fixed, the ending point must be equal to the location of the water faucet.

Our implementation of this constraint satisfaction problem utilizes the 360-ary tree data structure. The tree branches out for each vertical-horizontal turn, in which each child node represents a different turn degree and each node of the 360-ary tree tracks the corresponding coordinate after the turn. Thus, at least one of the leaf nodes should arrive at the outlet's coordinate. Each path from the root to a leaf represents one possible pipeline layout. Once the 360-ary tree is fully constructed, we simply scan all the leaf nodes to find the closest match(es) to the outlet's coordinates.

4.4 Ambiguity Elimination

For some tube-tube sequences, solving the constraint satisfaction problem in layout mapping generates multiple possible solutions. Consider the pipeline structure in Figure 12(a) and its measured pressure graph in Figure 12(b). Since there are two identical h-tube length segments, the constraint satisfaction generates 360 possible solutions (i.e., a v-h-turn can be 1 to 360-degrees), all of which satisfy the inlet/outlet positional constraints. Since most water pipelines are hidden on a flat wall plane, all but two solutions are likely. The two possible solutions (Figure 13) are that the pipeline travels on the left side or the right side of the wall.

The PipeProbe system resolves such ambiguities through additional mapping trips where listener devices are placed nearby the ambiguous paths obtained from the previous mapping trip. In Figure 13, a listener is attached to each wall location closest to each of the right and left paths. As Figure 13 shows, the ideal placements of the two listeners are at their mirror locations and further apart. Then, two listener devices listen in for packets broadcasted from the capsule as it travels by their locations. When the capsule flows nearby their locations, the listener device on the correct flow path receives many more packets than the listener device on the incorrect flow path. In other words, these two paths are disambiguated by the received packet rates of the two listener devices. Table 1 shows an experiment result to distinguish the ambiguity.

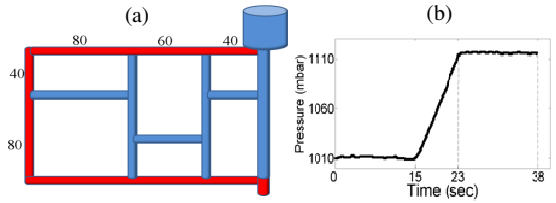

Figure 12. An example pipeline structure (a) and its pressure graph (b) produce two possible pipeline topologies shown in Figure 13.

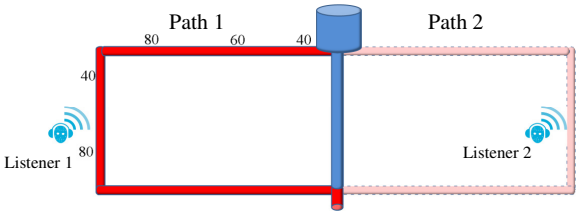

Figure 13. Two possible water pipeline topologies satisfy the starting position and ending position constraints and produced from the pressure graph in Figure 12(b).

Table 1. Received packet rate for distinguishing path ambiguity in the pipeline structure shown in Figure 13. The path receiving the higher received pack rate is the correct one.

	received packet # / total transmission packet #
Path 1	121/352
Path 2	13/352

5. Testbed

Figure 14 shows our pipeline testbed for evaluating the PipeProbe system. We purposely installed transparent pipe tubes (measuring 5cm in diameters) to enable direct observation on how well and consistent the PipeProbe capsule flowed inside of the pipeline as it went through the five prototype versions in the iterative design-test-analyze process. The testbed measures 18 cm x 140 cm x 345 cm with 51 transparent pipe tubes and 21 valves (with yellow or red handles) forming a pipeline network that has a 3x2 non-uniform grid on one vertical and two different horizontal travel paths on the ground. An input water source is attached to the plastic bin on the testbed's upper right corner. Thus, the upper right corner marks the starting point of all flow paths for our experiment. Figure 15 shows the length of each pipe tube.

Opening and closing different combinations of these 21 valves generates different flow paths with varying lengths and turn points for the traveling PipeProbe capsule. Figure 17 shows the 12 test scenarios in our evaluation. Each scenario was tested 6 times, i.e., the PipeProbe capsule makes six repeated mapping trips on the same flow path. Figure 15 marks the length of interconnecting pipe tubes and the positions of valves. For example, only opening all the top valves and all the left valves generates the simple flow path of test#1 (Figure 17) with 1 turning point and a traversal length of 320 centimeters. Four possible end points were installed in the test bed.

Data collection for each of the twelve test scenarios involved the following steps. First, the input water source was turned on to fill water tubes with water. Second, to produce a particular flow path, we set the valves accordingly. Third, the water faucet was opened to generate a continuous flow at a fixed rate. There are multiple ways to control the water flow rate. A simple method is to calculate the amount of time to consume N liters of water given a fixed input flow rate and pipe diameter. Note that only one faucet was opened at each time to generate a particular flow path. Fourth, we dropped the PipeProbe capsule into the water inlet. The PipeProbe capsule gathered and wirelessly transmitted sensor readings at a rate of 20 Hz while traveling inside the pipeline structure. Finally, the PipeProbe capsule was retrieved as it flowed out of the water outlet.

Figure 14. The experimental testbed for evaluating the PipeProbe system. 51 transparent tubes formed a 3-D non-uniform grid testbed. 21 valves with yellow handles were also installed. By opening and closing different valve combinations, different capsule flow paths and test scenarios were generated for evaluating the PipeProbe system.

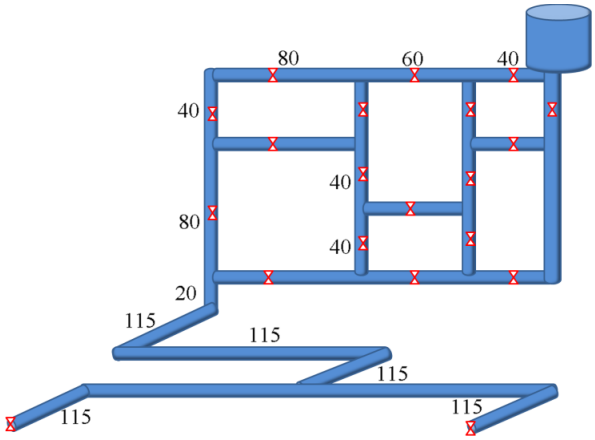

Figure 15. The lengths (cm) of 51 pipe tubes and the locations of 21 valves are drawn on the diagram.

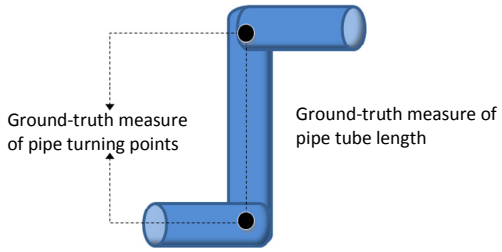

Figure 16. Ground-truth length and position measurements of pipe tubes and turning points.

6. Evaluation

Our main metric to evaluate the mapping accuracy of the PipeProbe system is defined as positional and length errors. Positional error is the Euclidean distance between the estimated coordinate and the ground-truth coordinate for each turning point on the flow path traversed by the PipeProbe capsule. Since the positional errors from previous estimation points carry into the error for subsequent estimation points, the positional error from a turning point is accumulative. Figure 16 shows the ground-truth coordinate for a turning point, which is measured as the midpoint of the turn. Length error is the difference between the estimated length and the ground-truth length of each pipeline tube on the flow path traversed by the PipeProbe capsule. Since the length of each pipeline tube is measured relative to its own starting point, the length error is non-accumulative. Figure 16 shows that the ground-truth length of a pipeline tube is measured from the midpoints of its two connecting pipe tubes.

6.1 Length Errors

Since the methods for deriving vertical tube length and horizontal tube length are different, we analyze horizontal and vertical tube length errors separately. Table 2 shows the number of measurements for each length of tube used in the 12 test scenarios for both horizontal and vertical flows. Each test scenario was tested 6 times, i.e., the PipeProbe capsule makes six repeated mapping trips on the same flow path. The average tube length from our test scenarios is 76-cm.

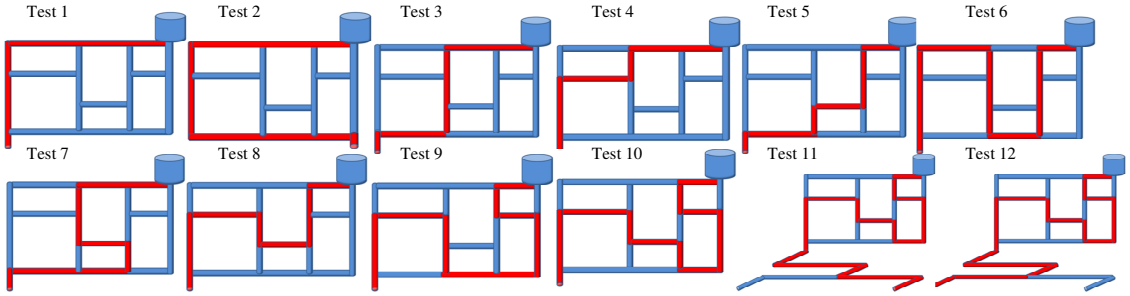

Figure 17. Flow paths (marked in red lines) in the 12 test scenarios. The estimated flow paths of the PipeProbe system are marked in red lines.

Table 2. The size of the collected dataset and the number of measurements categorized into vertical/horizontal tubes and various tube lengths.

	Actual ground-truth length (cm)	Number of measurements
Vertical tubes	20	24
	40	84
	80	48
	100	36
	120	24
	140	12
Horizontal tubes	40	84
	60	42
	80	54
	100	24
	115	60
	180	18

Figure 18 shows the cumulative density functions (CDFs) of the length errors for vertical tubes (the red line), horizontal tubes (the green line) and combined tubes (the blue line). The parametric settings were as follows: the flow velocity was 11.7 cm/second and the pressure sampling rate was 20 Hz. The dataset for the CDF is based on 510 length estimates for the pipeline tubes in the 12 test scenarios. The overall median length error was 2 centimeters, and 90% of the errors were less than 7 centimeters. The median length error for vertical-only tubes was 1 centimeter, and 90% of the errors were less than 4 centimeters. The median length error for horizontal-only tubes was 3 centimeters, and 90% of the errors were less than 7 centimeters. The test results demonstrate that our PipeProbe system achieves centimeter-level positional accuracy. Additionally, the estimation errors should be considered with respect to the 5-centimeter diameter, i.e., the error margin, of the pipe tubes within which the PipeProbe capsule flows.

Figure 19 depicts the average (standard deviation) of length errors for different pipe tube lengths, separating the vertical from horizontal pipe tubes. The dataset for the length errors was based on 228 length estimates for vertical pipe tubes and 282 length estimates for horizontal pipe tubes. The average (standard deviation) length error for vertical tubes, 1.5 cm (0.86 cm), was smaller than the average (standard deviation) length error for horizontal tubes, 3.6 cm (1.19 cm). This difference is due to the use of different techniques for calculating the lengths of horizontal and vertical tubes. When we calculate horizontal tubes, the error is accumulative, leading us to a less accurate result. Figure 19 shows that the average error in calculating a length generally increases with the length of the pipeline segment.

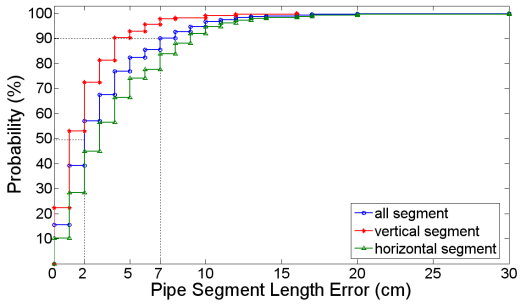

Figure 18. CDF of length errors.

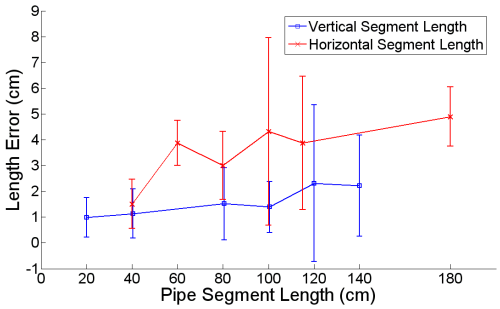

Figure 19. Average (standard deviation) length errors categorized into horizontal/vertical pipe segments and under different pipe segment lengths.

6.2 Positional Errors

Figure 20 shows the cumulative density function (CDF) for positional error. The dataset for the CDF was based on 588 positional estimates of pipeline turning points (i.e., v-turns and h-turns) in the twelve test scenarios. The median error was 6.8 centimeters, and 90% of the estimates had an error less than 15.8 centimeters.

Figure 21 plots the accumulated positional errors of turning points with respect to their traveled distances from

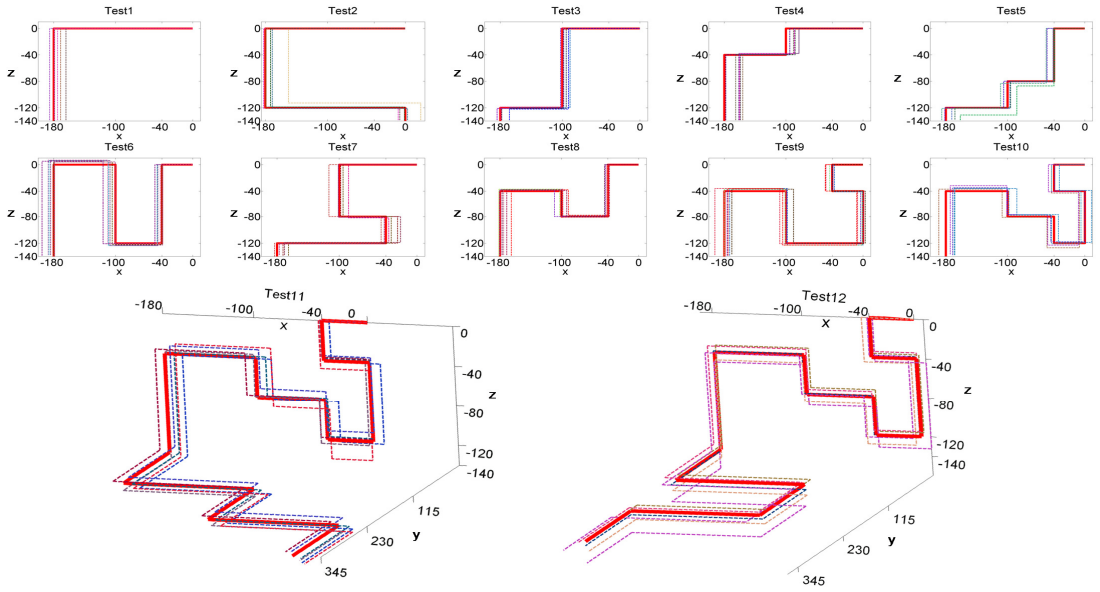

Figure 22. Estimated layouts over the actual traversal paths (in bold red lines) for all 12 test scenarios.

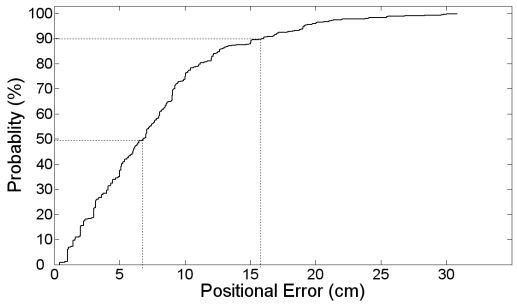

Figure 20. CDF of positional errors.

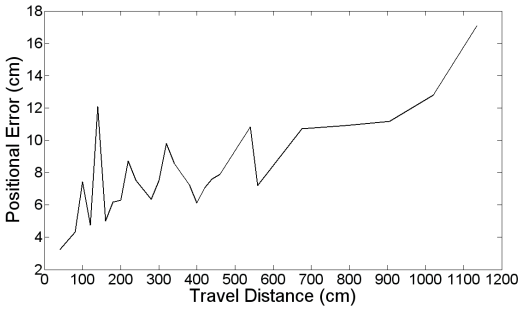

Figure 21. Positional errors under different distances traveled by the PipeProbe capsule.

the water inlet. The dataset for this plot was based on 588 positional estimates for pipeline turning points. The average traveled distance is 335 cm. The average error was 8.2 centimeters. The effect of error accumulation is evident – the average positional error increases with the distance the PipeProbe capsule travels. The effect of error cancellation is also present with occasional drops in average positional error with successive distance increments. Figure 22 illustrates the estimated layouts over the actual traversal paths for all 12 test scenarios.

6.3 Sampling Rate

The sample rates for pressure and gyroscope sensors are system parameters that directly affect positional error. We tested pressure and gyroscope sensors under different sample rates separately.

Figure 23 shows the position errors under different sampling rates [0.125Hz, 0.5Hz, 1Hz, 2Hz, 4Hz and 5Hz] for the pressure sensor. The dataset for the plot was based on one flow path from test#1 (Figure 17), measuring the positional errors. The analytical results show that a higher frequency rate generally decreases the positional error because the increased number of data samples enables a more accurate detection of the turning points. Figure 23 suggests that we should maintain the sample rate above 5Hz for water velocities under 11.74 cm/sec, since the positional error is only 4 centimeters.

Figure 24 shows the rotational angle calculated from angular velocities measured by the gyroscope sensor while the PipeProbe capsule is making a 90-degree right h-turn under different gyroscope sample rates [1.25Hz, 2.5Hz, 5Hz, 10Hz and 20Hz]. The results show that a higher frequency rate produces a more accurate rotational angle closer to the actual 90-degree turn. When the sampling rate drops down

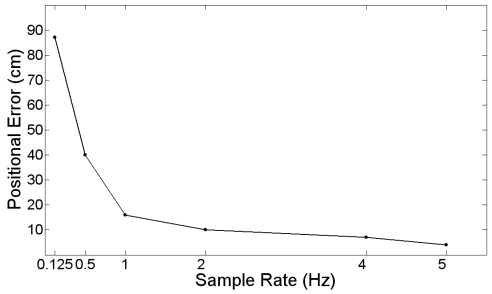

Figure 23. Positional errors under different sample rates for the pressure sensor.

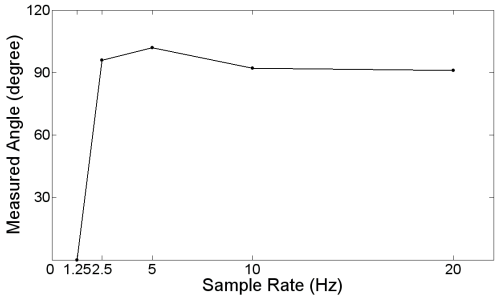

Figure 24. Rotation angle calculated from angular velocity measured by the gyroscope under different sample rates.

to 1.25 Hz, the calculated rotational angle becomes 0-degree and causes the turn detection to miss it completely. The reason is that the gyroscope did not collect enough samples at 1.25 Hz during the short amount of time the PipeProbe completes a turn.

6.4 Data Collection Trips

Figure 25 shows the positional errors of the PipeProbe system under different numbers of data collection trips. For example, six data collection trips mean that the PipeProbe capsule makes six repeated mapping trips on the same flow path. Then, the dataset is gathered from the 12 test scenarios and processed with statistical outlier removal and averaging to remove noise. The dataset for this plot was based on 588 positional estimates for pipeline turning points. The analytical results show that a higher number of mapping trips generally reduces the positional error and its standard deviation. Most likely, this is because an increased number of datasets enable more accurate reconstruction of the spatial topology of the pipelines. At one data collection trip, the PipeProbe system still achieved an average positional error of 4.9 centimeters and a standard deviation of 5.0 centimeters. At six data collection trips, the PipeProbe system achieved an average positional error of 4.1 centimeters and a small standard deviation of 2.2 centimeters.

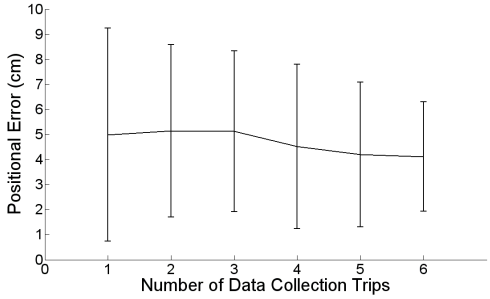

Figure 25. Average (standard deviation) positional errors under different numbers of data collection trips.

7. Discussion

In the PipeProbe system, there two assumptions: (1) all pipelines have the same diameter, and (2) the position of inlet/outlet point is known. Here we discuss how to relax these assumptions. In addition, we also address the method to reduce the size of PipeProbe.

A change in the internal pipe diameter causes a corresponding change in the volumetric flow rate and velocity. To detect different water flow velocities, we will augment PipeProbe with an extra paddlewheel speed sensor to measuring its flow velocity directly. This would also avoid the need to create constant water flow velocity in our current system. The paddlewheel speed sensor works as follows. As water flow causes the paddlewheel to spin, the magnets imbedded in the paddle spin produce electrical pulses proportional to its flow velocity.

There are some cheap and handy tools that architects use on a daily basis to measure the 3D position of inlet/outlet points. For example, barometer can measure building height. Laser rangefinders can measure width/length. Higher-end meters generally provide more accurate measurements.

There are several ways to reduce the size of the PipeProbe such that it can fit into most pipes. For example, the current PipeProbe has a loose packaging and does not fully utilize all its internal space. The largest component in the PipeProbe is the Eco mote whose size is 13 mm (L) × 11 mm (W) × 7 mm (H). Therefore, we can shrink PipeProbe by custom-making a spherical shell at the mm scale. Making a custom printed circuit board will also eliminate most wiring that takes up space. We are currently working on the next version of PipeProbe with a size reduction from 4-cm diameter to 2-cm diameter.

8. Related Work

Recent projects that use wireless sensor network technologies for measuring water flow and detecting leakage include the NAWMS project [2] and the PIPENET project [1]. The NAWMS project provides information about where and how much water people are using by attaching vibration sensors to pipe surfaces. NAWMS is easy to install, but is not very feasible since all of the pipes in a building have be

installed with a sensor. This gets expensive when the pipeline structure is complex. Similarly, the PIPENET project monitors water flow and detects leaks by attaching acoustic and vibration sensors to external pipelines and pressure sensors to internal pipelines. In contrast with these projects, the PipeProbe system does not assume that water pipes are accessible for someone to attach sensor modules to them.

Alvarado et al. [9] developed a robotic fish under a foot long that closely mimics a real fish's natural swimming motion. This robofish is equipped with sensors to detect environmental pollutants. However, it is considerably larger than the PipeProbe capsule, and requires a 2.5-5W external power source to run a motor.

There are also some noteworthy projects that emphasize fixture classification of water pipeline. Fogarty et al. [7] used a microphone to monitor the plumbing system and infer water usage within a household. Nevertheless, microphone based recognition is obstructed by ambient noise. The recently proposed HydroSense [3] is a promising system which uses the pressure fingerprint of each water fixture to identify its activity within a building accurately. It uses single-point detection and exploits the "water hammer" effect, which is uniquely produced by every fixture. By detecting and identifying the fixture's fingerprint, it can infer if the fixture is on or off. This project strengthens the case for processing pressure signals, which are stable and uninhibited by ambient noise

With regards to water flow estimation and fixture identification, we are not aware of any prior work using a sensor probe for mapping the pipeline structures throughout a home. We consider the other systems complementary to our approach because PipeProbe maps water pipes to assist in locating the leakage and monitoring the pipes.

Monitoring a house's infrastructure provides behavioral information about its inhabitants. Patel et al. [8] identified particular devices by detecting the electrical noise caused by the devices' operation. The recently proposed ViridiScope [5] uses a combination of the magnetic field, acoustic information, and light intensity to estimate the power consumption within a household.

9. Conclusion

The proposed PipeProbe system presents a novel mobile sensor system for determining the spatial topology of hidden water pipelines. Experimental results from our testbed achieved a median length error of 2 centimeters, and 90% of the tests had a length error of 7 centimeters or less while estimating the lengths of pipe tubes. We had a median positional error of 6.8 centimeters, and 90% of the tests had a positional error of 15.8 centimeters or less while estimating the pipe's turning points. By using a tiny capsule to sense pressure readings as it traverses through the pipelines, the PipeProbe system produces accurate mapping. Additionally, PipeProbe is highly energy-efficient, since its physical movement leverages the existing water flow.

10. References

[1] Stoianov, I., Nachman, L., Madden, S., and Tokmouline, T. PIPENET: A Wireless Sensor network for pipeline monitoring. *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks* (2007), 264-273.

[2] Kim, Y., Schmid, T., Charbiwala, Z.M., Friedman, J. and Srivastava, M.B. NAWMS: Non-Intrusive Autonomous Water Monitoring System. *Proceedings of the ACM Conference on Embedded Network Sensor Systems SenSys* (2008), 309-322.

[3] Froehlich, J., Larson, E., Campbell, T., Haggerty, C., Fogarty, J. and Patel, S. HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity. *Proceedings of the international conference on Ubiquitous computing* (2009), 235-244.

[4] Park, C. and Chou, P.H. Eco: Ultra-Wearable and Expandable Wireless Sensor Platform. *Proceedings of the International Workshop on Body Sensor Networks* (2006), 162-165.

[5] Kim, Y., Schmid, T., Charbiwala, Z.M. and Srivastava, M.B. ViridiScope: Design and Implementation of a Fine Grained Power Monitoring System for Homes. *Proceedings of the 11th international conference on Ubiquitous Computing* (2009), 245-254.

[6] The STMicroelectronics LISY300AL gyroscope chip http://www.st.com/stonline/books/pdf/docs/14753.pdf

[7] Fogarty, J., Au, C., and Hudson, S.E. Sensing from the basement: A feasibility study of unobtrusive and low-cost home activity recognition. *Proceedings of the Annual ACM Symposium on User Interface Software and Technology* (2006), 91-100.

[8] Patel, S. N., Robertson, T., Kientz, J. A., Reynolds, M. S. and Abowd, G. D. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. *Proceedings of the international conference on Ubiquitous computing* (2007), 271-288.

[9] Alvarado, P. V. and Youcef-Toumi, K. Performance of Machines with Flexible Bodies Designed for Biomimetic Locomotion in Liquid Environments. *ICRA* (2005), 3324-3329.

[10] PPFA (Plastic Pipe and Fittings Association), Design Guide, Residential PEX Water Supply Plumbing Systems (2006), http://www.toolbase.org/PDF/DesignGuides/pex_designguide.pdf.

[11] American Society of Home Inspectors, http://www.ashi.org/media/press/.

[12] Dump, C. Principles of Home Inspection: Plumbing. *Dearborn Home Inspection Education* (2003).

[13] MS5541C Pressure Sensor http://www.intersema.ch/products/guide/calibrated/ms5541