A Low-Cost Method for Measuring Surface Currents and Modeling Drifting Objects

Huang-Chen Lee, *Student Member, IEEE*, Chun-Yu Lin, Chun-Han Lin, *Student Member, IEEE*, Sheng-Wen Hsu, and Chung-Ta King, *Senior Member, IEEE*

***Abstract*—The ability to measure and model water currents is essential to ensure the safety and correct operations of many water surface activities. For example, the complex currents in harbors seriously affect the safety of vessels, while river turbulence and vortices cause safety hazards for people participating in white-water rafting, swimming, and other sports activities. In this paper, we demonstrate a low-cost method to measure and model surface currents. Specially designed GPS sensors are dropped into the water to record their drifting trajectories, which are then transformed into a current map to show the characteristics of the currents, including their velocity and direction. A proof-of-concept experiment shows the feasibility of the proposed method and how turbulence locations can be identified. We further demonstrate that the derived current map can be used to construct a mobility model of a drifting object and generate its virtual drifting trajectories. Our analyses show that the generated virtual trajectories closely fit the collected trajectories.**

***Index Terms*—Current measurement, drifting trajectory, GPS sensor, mobiltiy model, modeling, surface current, water current.**

I. Introduction

MEASURING currents, including ocean and river currents, plays a vital role in the safety management of harbors and rivers. For example, the complex currents caused by harbor structures and docks often pose threats and safety hazards to vessels navigating through busy harbors and narrow channels. An up-to-date *current map* should thus help the vessels in avoiding these hazardous areas. This information is also invaluable in water sports such as whitewater rafting and swimming, as a current map of a river could be used as an essential guide to avoid vortexes or turbulences and thus improve safety. An affordable and easy-to-deploy method to measure currents is therefore greatly needed.

Many sensing techniques have been developed to measure surface currents. For example, the *acoustic Doppler current profiler* (ADCP) [1] can be installed on seabeds/riverbeds to measure the velocity and direction of currents in a region. It emits ultrasonic pulses that scatter moving particles in the water to determine their velocity according to the signals that bounced back. ADCPs have been deployed to measure currents [2]–[6] with high accuracy and reliability.

One limitation of ADCPs is their fixed position and limited range of monitoring. The maximum profiling range of commercial available ADCPs is about 300 m in a cross section. To obtain a comprehensive 2-D current map over a wide region, a large number of ADCPs have to be deployed. Each commercial ADCP costs around 16 000 to 100 000 USD, plus underwater deployment, cabling, and maintenance costs. The high cost limits ADCP use to mainly scientific research.

Another approach to surface current measurement is to use remote sensing such as *interferometric synthetic aperture radar* (InSAR) [7]–[11]. InSAR uses radar images from aircraft or satellites; hence, the monitoring area can be very large, e.g., 5 km $\times$ 5 km. The state-of-the-art InSAR satellite TerraSAR-X [9] can provide up to 1 m of spatial resolution when measuring the velocity of a surface current with an error of 10 cm/sec. However, InSAR requires expensive facilities such as satellites and airplanes to obtain images. Processing and accessing the images are also very involved, and it is difficult for local authorities to obtain the required information on a regular basis using nonprofessional employees.

In this paper, we demonstrate a low-cost method to measure the currents in a region. The system can easily be operated by nonexpert personnel to obtain a current map on a regular basis or as needed. It can serve as a stand-alone or as a complement to the more expensive systems described earlier.

The workflow of the proposed method is shown in Fig. 1. This project is an extension of our previous work [12], which used a special wireless GPS sensor that could float on water instead of using conventional stationary sensors. Several such sensors are dropped into the water, drifting passively with the water currents and recording their trajectories. The recorded data are collected and used to build a *parameter map* of the water way, which is a 2-D current map that shows important parameters, such as current velocity and direction, of the currents at different locations.

An example parameter map is given in Fig. 2, showing the directions of the currents in a harbor entrance. The region of interest has been divided into rectangular areas, and the arrows indicate flow direction. Such a map can have many uses. For example, as mentioned earlier, it can be used by harbor authority or vessel pilots to optimize the flow of ships and to

Manuscript received January 16, 2010; revised June 23, 2010; accepted June 24, 2010. This work was supported in part by the National Science Council, Taiwan, under Grant NSC 97-2218-E-007-001, by the Ministry of Economic Affairs, Taiwan, under Grant 96-EC-17-A-04-S1-044, and by Microsoft Research (SensorMap: Browsing the Physical World in Real-Time 2007 Awards). The Associate Editor coordinating the review process for this paper was Dr. Jesús Ureña.

H.-C. Lee, C.-Y. Lin, C.-H. Lin, and C.-T. King are with the National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: huclee@mx.nthu.edu.tw).

S.-W. Hsu is with MediaTek, Hsinchu 300, Taiwan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2010.2062730

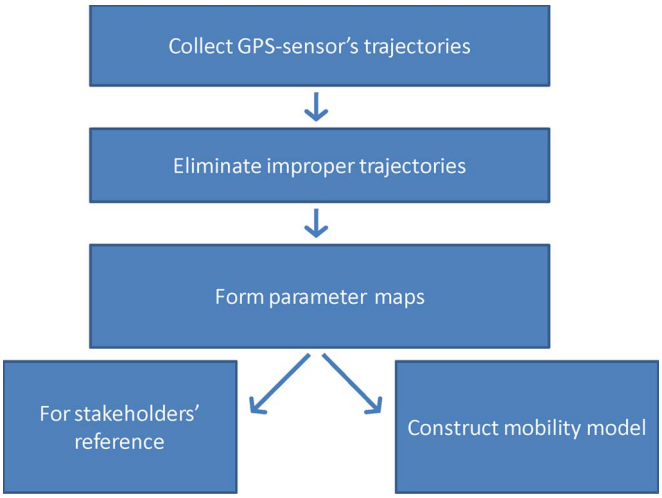


Fig. 1. Workflow of the proposed method.

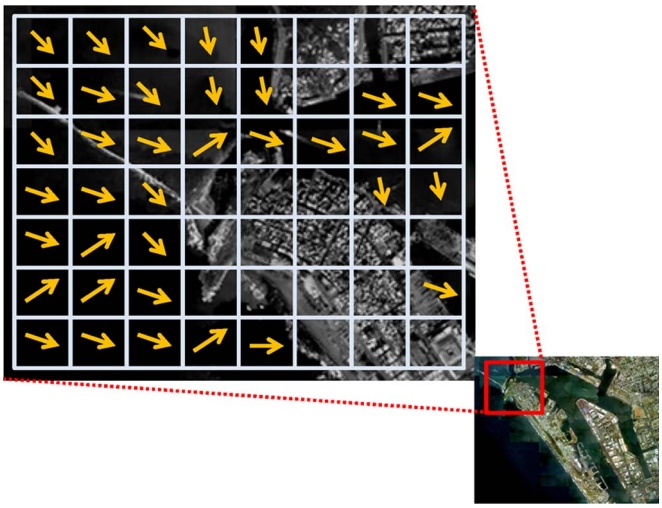


Fig. 2. Parameter map showing current directions in a harbor entrance. The region of the harbor entrance is divided into rectangular areas.

ensure navigational safety. We will discuss this in further detail in the following sections.

Another important application of the parameter map is to derive a mobility model of drifting objects in the region of interest. The mobility model can be used to simulate the possible moving trajectories of floating objects in the region [13]–[15], which, in turn, can be used for various studies, for example, how drifting logs, ice chunks, or even garbage move along a water way and how these objects behave in the presence of different safety measures.

A comparison of the proposed method and the existing current measurement techniques is shown in Table I. In summary, our proposed method has the following advantages: 1) it makes it possible for nonspecialized parties to drop and get information about the currents in a target region; 2) the cost of the system is low; each prototype wireless GPS-sensor costs less than 60 USD; and 3) the proposed method gives a comprehensive 2-D current map without a large-scale deployment.

II. Collecting Trajectories

The main device used in our proposed system is a special GPS sensor. Fig. 3 shows this device and its block diagram. It is encapsulated to be waterproof and floats on the water. The device is based on *Telos* [21], an ultralow-power wireless sensor platform based on MSP430 [22]. It is integrated with an MediaTek GPS module FPGMMOPA1 [23]. The errors of the GPS module in terms of position, velocity, and timing are less than 3 m, 5 cm/s, and 100 ns, respectively. Issues regarding GPS accuracy will be discussed later in this paper. This sensor device is programmed to record its latitude and longitude into internal storage every second. In addition, it is also equipped with an onboard 2.4-GHz radio transceiver, i.e., CC2420, which allows it to wirelessly report its location to shore-based receivers in real time.

Some modifications have been made to the sensor platform. We modified the USB interface IC (FT232BM) and the GPS module so that they share the same UART interface of the MSP430. The GPS module was set to output an NMEA 0183 string [24] every second to the MSP430 to report the current location. To reduce the data storage required, the MSP430 extracts the location, velocity, and direction from the NMEA string "GPRMC" and converts them into a 9 bytes abstracted data structure. These location data are logged into the nonvolatile internal memory (M25P80) of Telos for later downloading.

To prove the concept, we conducted an experiment over an accessible segment of a river. This segment is within latitude $[\mathrm{N}24^\circ 49.255' \sim \mathrm{N}24^\circ 49.31']$ and longitude $[\mathrm{E}121^\circ 00.070' \sim \mathrm{E}121^\circ 00.082']$, located in Hsinchu, Taiwan, which amounts to an area about 10–15 m wide and 100 m long. Fig. 4 shows this area. The reason for choosing this river segment is that it is accessible and controllable without using boats.

The experiment was conducted by dropping the sensors upstream and retrieving them downstream. This procedure was repeated several times. A total of 45 trajectories were collected, each consisting of a series of *points* with parameters in velocity and direction. A *point* also stands for the runtime in 1 s of a trajectory. There are about 7000 points in the 45 trajectories.

III. Analysis and Building Parameter Maps

The collected trajectories were processed to filter out improper ones, e.g., those in which the GPS module lost its GPS signal or the sensors were trapped by stones. Among the 45 collected trajectories, 15 were eliminated. The remaining 30 trajectories contained 4557 points, which are shown in the scatter plot in Fig. 5.

The distributions of velocity and direction of the collected trajectories are shown in Figs. 6 and 7, respectively. In Fig. 6, the *cumulative probability function* (CDF) of velocity follows a general Pareto distribution with a shape parameter (k) of 0.031 and a scale parameter (*sigma*) of 0.75. We denote this Pareto distribution as $Dist_{\mathrm{velocity}}$ and refer to it as the *velocity distribution*. The velocity can be measured as a unit of nautical miles per hour or *knots*, where 1 nmi is 1852 m and 1 knot is about 0.5 m/s. In Fig. 6, we can see that, in 70.17% of the running time, the velocity is smaller than 1 knot, and in 97.65% of the time, it measures less than 2 knots.

TABLE I
COMPARISON OF CURRENT MEASUREMENT TECHNIQUES

	Equipment cost	Measurement range	Long-term measurement	Target applications
Proposed method	Low	Small to median	Yes	Water activity safety Harbor and channel control Drifting object tracking Scientific research
ADCP	Median	Small	Yes	Channel/harbor safety Scientific research
InSAR	Very high	Very large	No	Scientific research

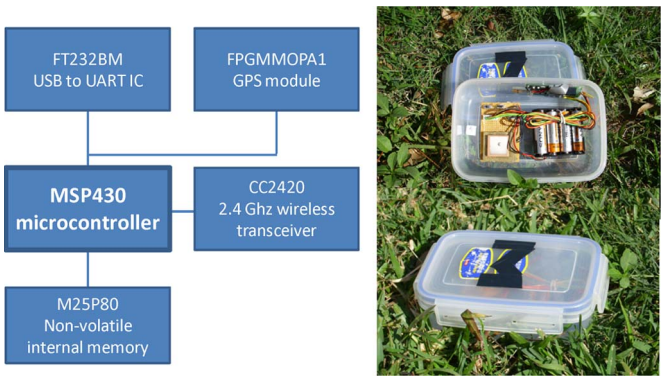


Fig. 3. Wireless GPS sensor used in the study.

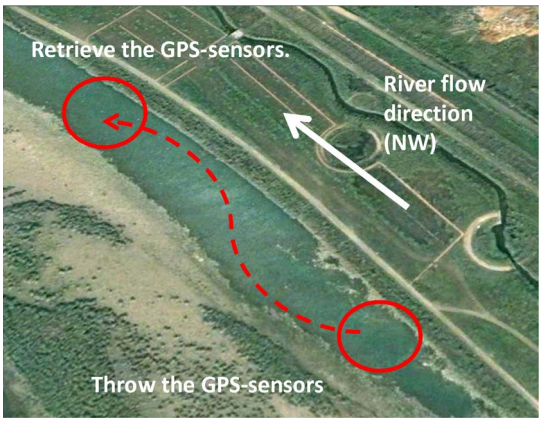


Fig. 4. Bird's-eye view of the area for the proof-of-concept experiments. The GPS sensors were dropped at the lower right (upstream) and retrieved at the upper left (downstream).

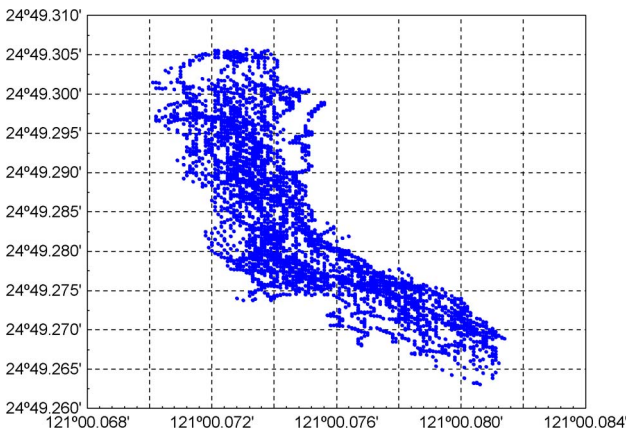


Fig. 5. Scatter plot of 4557 points from 30 collected trajectories.

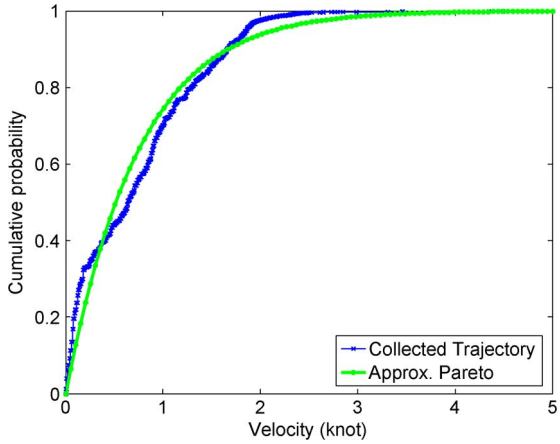


Fig. 6. CDF of the current velocity ($Dist_{\text{velocity}}$).

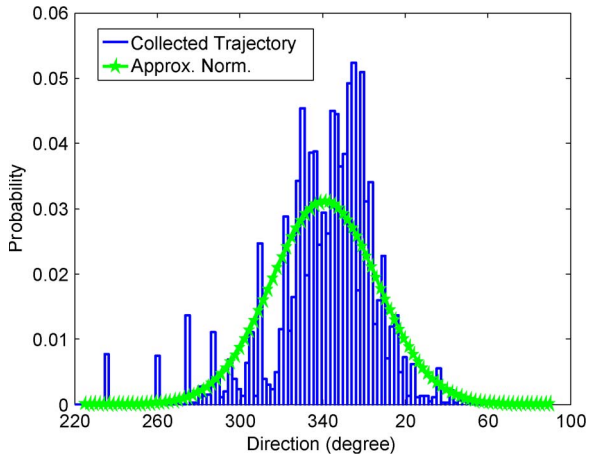


Fig. 7. Probability distribution of the current direction ($Dist_{\text{direction}}$).

Fig. 7 shows the distribution of the directional data. The directional data were retrieved from the NMEA string "GPMRC" from the GPS module. It indicates the *course over ground* degrees from the true north (i.e., 0° to the true north). The probability distribution generally follows a normal distribution with a mean of 340.97° and a standard deviation of 25.67°. This normal distribution is denoted as $Dist_{\text{direction}}$.

In the following, we discuss how to transform the collected trajectories into *parameter maps*. First, we enclose the area of interest with a rectangle, denoted as S_i, and divide it into $m \times n$ cells. Each cell is denoted as $C_{x,y}$, where $1 \leq x \leq m$ and $1 \leq y \leq n$, i.e.,

$$S_i = \{C_{1,1}, C_{1,2}, \ldots, C_{1,n-1}, C_{1,n}, C_{2,1}, C_{2,2}, \ldots, C_{m,n-1}, C_{m,n}\}. \quad (1)$$

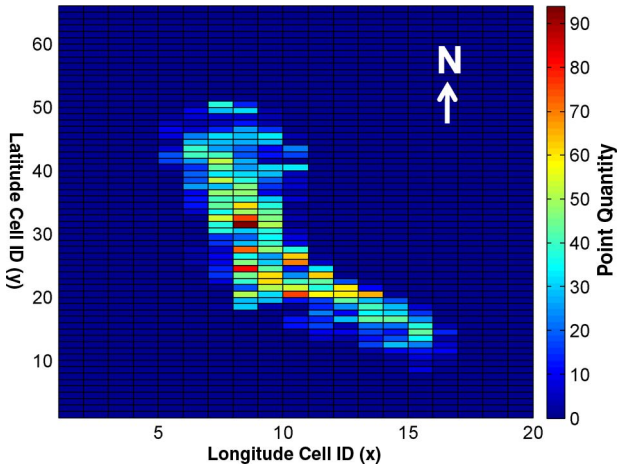


Fig. 8. Point quantity map (N *Pmap*) from our proof-of-concept experiment.

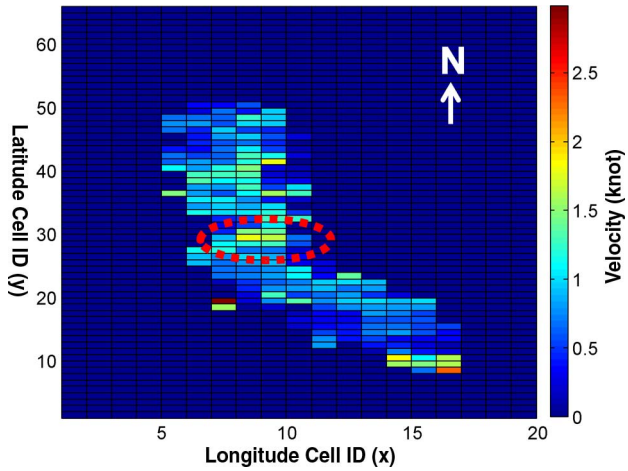


Fig. 9. General velocity map (V *Pmap*) from our proof-of-concept experiment.

The cell size should properly be chosen to include a sufficient number of location points. In our proof-of-concept experiment, we chose a cell size of 0.001 $\times$ 0.001 in longitude and latitude, which is approximately 1.852 m $\times$ 1.852 m. S_i is divided into 20 $\times$ 65 cells, as shown in Fig. 8.

Let $T_i = \{t_1, t_2, t_3, \ldots\}$ denote the set of trajectories $t_1, t_2, t_3, \ldots$ collected from area S_i. A trajectory t_j consists of a series of points $p_1, p_2, p_3, \ldots$, where each point p_k holds its longitude, its latitude, and some parameters such as velocity and direction. A point stands for the location and movement of the floating object in a single second. These are given as follows:

$$T_i = \{t_1, t_2, t_3, \ldots\}. \tag{2}$$

$$t_j = \{p_1, p_2, p_3, \ldots\} \tag{3}$$

$$p_k = (longitude(p_k), latitude(p_k), velocity(p_k), direction(p_k)). \tag{4}$$

From the trajectories and cells, we then determine whether a point p is inside a cell $C_{x,y}$. The set of points inside a cell $C_{x,y}$ is collected and denoted as $P_{x,y}$. Let $N_{x,y} = |P_{x,y}|$ be the number of points in the cell.

From all the points inside a cell $C_{x,y}$, we next calculate the *point quantity* $N_{x,y}$, *general velocity* $V_{x,y}$, and *general direction* $D_{x,y}$ of the cell. $V_{x,y}$ and $D_{x,y}$ are the *median value* of velocity and direction in $P_{x,y}$. Note that the median value is used instead of the mean value because the variation of velocity and direction of the points within a cell may be high if there are too few points in the cell, e.g., less than five points; the median value is more statistically robust. The calculations are given as follows:

$$\begin{aligned}
&\text{For a specific cell } C_{x,y} \in S_i: \\
&P_{x,y} = \{\forall p \in C_{x,y}\} \\
&N_{x,y} = |P_{x,y}| \\
&V_{x,y} = median\left(velocity(p_1), velocity(p_2), \ldots\right) (\forall p \in P_{x,y}) \\
&D_{x,y} = median\left(direction(p_1), direction(p_2), \ldots\right) \\
&\qquad (\forall p \in P_{x,y}).
\end{aligned} \tag{5}$$

After calculating $N_{x,y}$, $V_{x,y}$, and $D_{x,y}$ for all $C_{x,y}$ in S_i, we can now build the parameter maps for S_i. For example, from our proof-of-concept experiment, we can obtain the parameter map of point quantity shown in Fig. 8. Note that the parameter map is not scaled to the dimension of the real geographic area. The point quantity map, denoted as N *Pmap*, could be used to determine the probability that drifting objects might traverse through a cell. This information is very useful for building the mobility model, as discussed in the following sections.

Fig. 9 presents the parameter map of the general current velocity, hereafter denoted as V *Pmap*, obtained from our experiment. We can see that the velocities in the central portion of the river (enclosed by the dotted red circle) are higher than in the surrounding cells. This cautions harbor managers and vessel pilots to be more careful in this area.

Fig. 10 shows the parameter map of the general current direction, hereafter denoted as D *Pmap*. In the bottom-right portion of this map (enclosed by the dotted green circle), we can see that the directions of the currents consistently point to the upper left portion of the map. From the map, we can also indentify two areas of turbulence, enclosed by the two dotted red circles, which should be noted for navigational safety.

The parameter maps have various uses, depending on their applications. Even when applied to safety management, different applications may have different requirements. For example, while parameter maps provide essential information for applications to use, a turbulence of 5 $\times$ 5 m may pose a serious threat to whitewater rafting but may not affect cargo ships. In the next section, we will examine one application in more detail, i.e., the derivation of a mobility model of drifting objects in the area of interest.

IV. Mobility Model of Drifting Objects

Mobility models have widely been used in research areas such as mobile networks, transportation, and disease dissemination. In mobile wireless networks, research on network routing and deployment often requires appropriate mobility models to evaluate performance. Unfortunately, most existing mobility models focus on objects moving on solid ground. With an increasing interest in deploying wireless sensor nodes on fluid flows for natural disaster prevention and environmental protection [13]–[15], the creation of mobility models of drifting objects on currents is imperative.

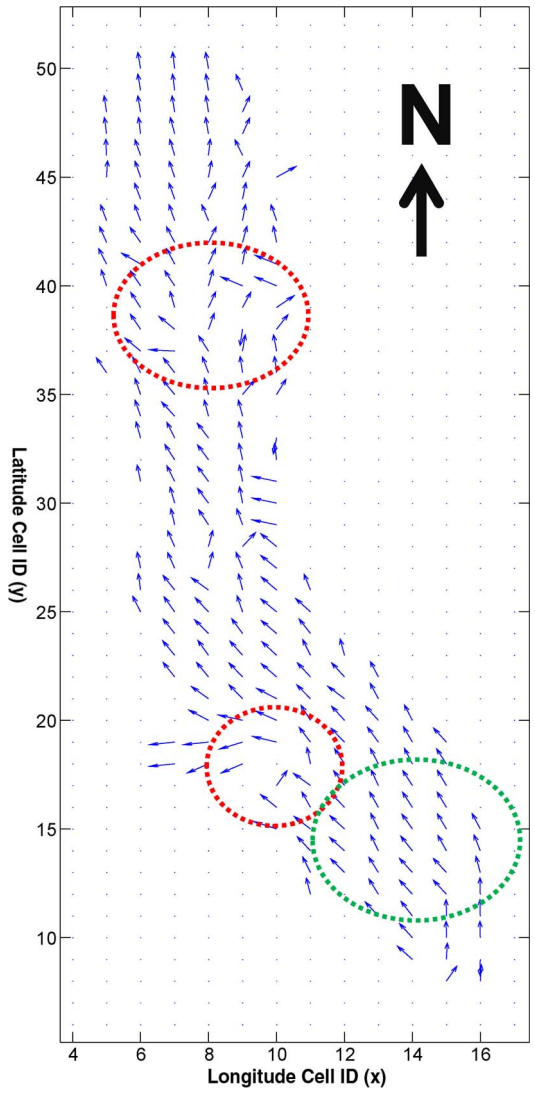


Fig. 10. General direction map (D *Pmap*) from our proof-of-concept experiment.

Existing mobility models can be categorized as synthetic and trace based. Synthetic approaches generate virtual trajectories without referring to field information. Instead, mobility is modeled purely on mathematical definitions, and thus, it is difficult to capture real situations. On the other hand, trace-based approaches use information collected in real environments as the basis for the mobility model.

One of the most popular synthetic mobility models is the random waypoint model [16]. The movements of a node are modeled following a walk-and-pause cycle. The destination and moving speeds are randomly chosen. The model is simple and easy to use for numerical analysis and programming in simulators. In [17], a mobility model considering obstacles is proposed, in which a *Voronoi* diagram is used to determine the paths of simulated nodes. Another well-known mobility model is the *reference point group mobility* model [18], which assumes that nodes in the same group have similar behaviors and that the path of a group follows predefined checkpoints of the group's logical center. In addition, the members of the group randomly move around the logical center. These synthesis mobility models generate nodes' movements using algorithmic or mathematical formulations based on intuition and not on real-world behaviors.

Trace-based mobility models generate the moving behaviors, e.g., the virtual trajectories, of nodes based on information collected in the field. For example, the wireless local area network (WLAN) mobility model [19] is based on the accessing log of WLANs on the Swiss Federal Institute of Technology Zurich (ETH Zurich) campus. In [20], a 13-month history log of network associations is examined to acquire the movement characteristics of Wi-Fi devices. The mobility models obtained by the trace-based approaches are akin to real-world behavior, although they may be confined to the environment where the data are collected. In this paper, we follow the trace-based approach to build a mobility model for drifting objects. As mentioned in Section I, such a mobility model can be used to study how drifting logs, ice chunks, or even garbage may move along a waterway and how they behave under different safety measures.

Given the parameter maps N *Pmap*, V *Pmap*, and D *Pmap*, we discuss how to construct the mobility model of drifting objects. The mobility model can be utilized as a function to generate the movements of an object in time and spatial domains. The movements may be expressed as a virtual trajectory generated from a stochastic computation point by point. The pseudocode of the proposed trajectory generation algorithm and the block diagram are shown in Fig. 11.

The trajectory generation function generates a trajectory in S_i starting from some points in INITIAL_LOCATION (line 04). The initial point is randomly selected from within INITIAL_LOCATION to enhance the variability of the virtual trajectories. In our proof-of-concept experiment, INITIAL_LOCATION is located at $[\mathrm{E}121°00.080' \pm 0.001', \mathrm{N}24°49.265' \pm 0.003']$. The points in the virtual trajectory are then recursively generated point by point. For example, if the present point is p'_k and $p'_k \in C_{x1,y1}$, the direction and velocity of the next point p'_{k+1} is generated using the parameters $V_{x1,y1}$ and $D_{x1,y1}$ of cell $C_{x1,y1}$. The function **get_cell_id()** (line 05) refers to *current_location* and returns the cell ID $C_{x,y}$, where the $current\_location \in C_{x,y}$. The points are sequentially generated until the runtime is longer than MAX_TIME or *current_location* is in GOAL. For our experiment, MAX_TIME $= 450$, and GOAL $= [\forall\, location | latitude > \mathrm{N}24°49.304']$.

The function **isvalid()** (line 08 and 12) returns FALSE if the input point p' is in $C_{x,y}$ and the corresponding $N_{x,y} = 0$, indicating that the cell is an *empty cell*. We define the set of empty cells as a *boundary cell set* $C_{\mathrm{boundary}} = \{C_{x,y} | \forall\, C_{x,y} \in S_i \wedge N_{x,y} = 0\}$, so the virtual trajectories are constrained in the cell set $\{S_i - C_{\mathrm{boundary}}\}$. Without boundary constraints, the generated trajectories might enter locations the collected trajectories never visited, e.g., shores or obstacles. The function **generate_movement()** (line 7 and 28) calculates each single movement according to $V_{x,y}$ and $D_{x,y}$ and the current location $p' \in C_{x,y}$.

The function **generate_velocity()** (line 21) uses the input $V_{x,y}$ to randomly generate the velocity of the next movement from the velocity Pareto distribution, i.e., $Dist_{\mathrm{velocity}}$, and the **generate_direction()** (line 22) uses $D_{x,y}$ to generate the direction of the next movement from the direction normal distribution $Dist_{\mathrm{direction}}$. (Note that $Dist_{\mathrm{velocity}}$ and $Dist_{\mathrm{direction}}$ are

```
Function Trajectory_Generation
01 load the parameter maps (N Pmap, V Pmap and D Pmap) of S_i
02 current_location = INITIAL_LOCATION
03 new_trajectory = 0
04 while ( |new_trajectory| < MAX_TIME or current_location in GOAL)
05   C_x,y := get_cell_id( current_location )
06   V_x,y, D_x,y := get_cell_parameter(C_x,y)
07   movement := generate_movement(V_x,y, D_x,y)
08   if( isvalid (current_location + movement) = TRUE)
09     current_location:= current_location + movement
10   else
11     bounce_movement := get_bounce(V_x,y, C_x,y)
12     while( isvalid(current_location + bounce_movement) != TRUE)
13       bounce_movement := get_bounce(V_x,y, C_x,y)
14     end while
15     current_location := current_location + bounce_movement
16   end if
17   new_trajectory := append(current_location)
18 end while
19 output new_trajectory;
20 function generate_movement(V_x,y, D_x,y)
21   cell_velocity := generate_velocity(V_x,y)
22   cell_direction := generate_direction(D_x,y)
23   new_movement := cell_velocity × cell_direction
24   return new_movement
25 end function
26 function get_bounce(V_x,y, C_x,y)
27   bounce_direction := compute the bounding movement direction using
     C_x,y)
28   new_movement := generate_movement(V_x,y, bounce_direction)
29   return new_movement
30 end function
```

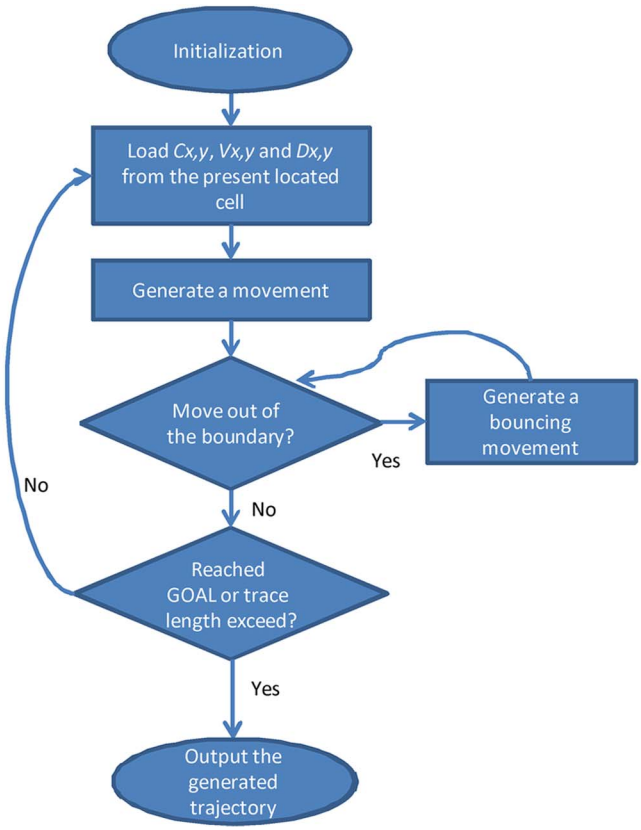


Fig. 11. Pseudocode of the proposed trajectory generation algorithm and the flow chart.

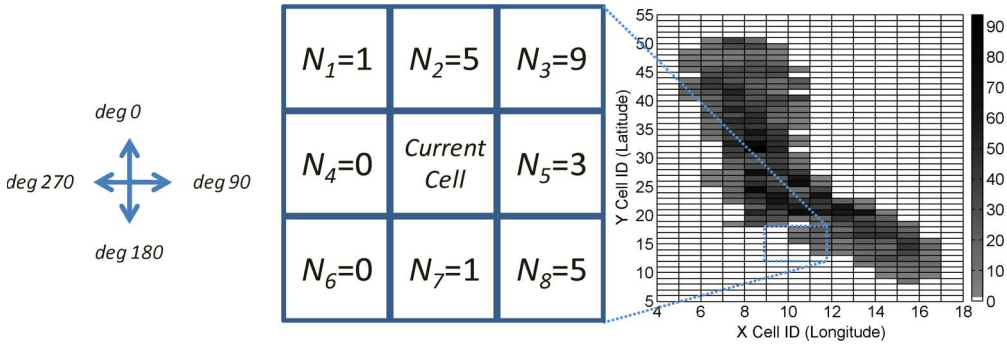


Fig. 12. Point quantities of the surrounding cells of *CurrentCell* are used to determine the probabilities of the direction of the bouncing movement.

based on the collected trajectories of S_i.) In lines 21–22, the velocity and direction of the next movements are obtained by randomly sampling a value from their corresponding distributions. Line 23 generates the 2-D movement in the target space S_i.

The function **get_bounce()** (lines 11 and 13) defines how a bouncing movement is generated. A drifting object on water will bounce when it hits certain obstacles such as stones or the shore. For example, if the movement m' generated by **generate_movement()** (line 07) causes the next point $p'_{k+1} = p'_k + m'$ to fall into an empty cell $C_{\text{empty}} \in C_{\text{boundary}}$, then m' must be discarded and **get_bounce()** is used to generate a bouncing movement. The bouncing direction is calculated based on the idea that cells with a higher point quantity N have a higher probability of being visited. Therefore, the bouncing movement has a higher probability of moving into the direction of the surrounding cell, which holds more points.

Fig. 12 shows an example describing how the bouncing movement direction is determined. In this example, the present point p is within *CurrentCell*. The point quantities of the *surrounding cells* of *CurrentCel* are indicated by $N_1, N_2, \ldots, N_8$, respectively. The sum of the point quantity of the surrounding cells is $N_{\text{sum}} = N_1 + N_2 + N_3 + \cdots + N_8 = 24$. The probability of bouncing direction degree of 0 is denoted by $P(0^\circ)$, which is given by $P(0^\circ) = N_2/N_{\text{sum}} = 5/24 = 0.208$. The probabilities of other bouncing directions can similarly be computed. Next, the direction of the bouncing movement can randomly be sampled from these probabilities.

The proposed trajectory generating function was evaluated using Matlab [25]. Three virtual trajectories using the parameter maps from our experiment are presented in Fig. 13. Note that the generated virtual trajectories look very much like the collected real trajectories. This was further evaluated using

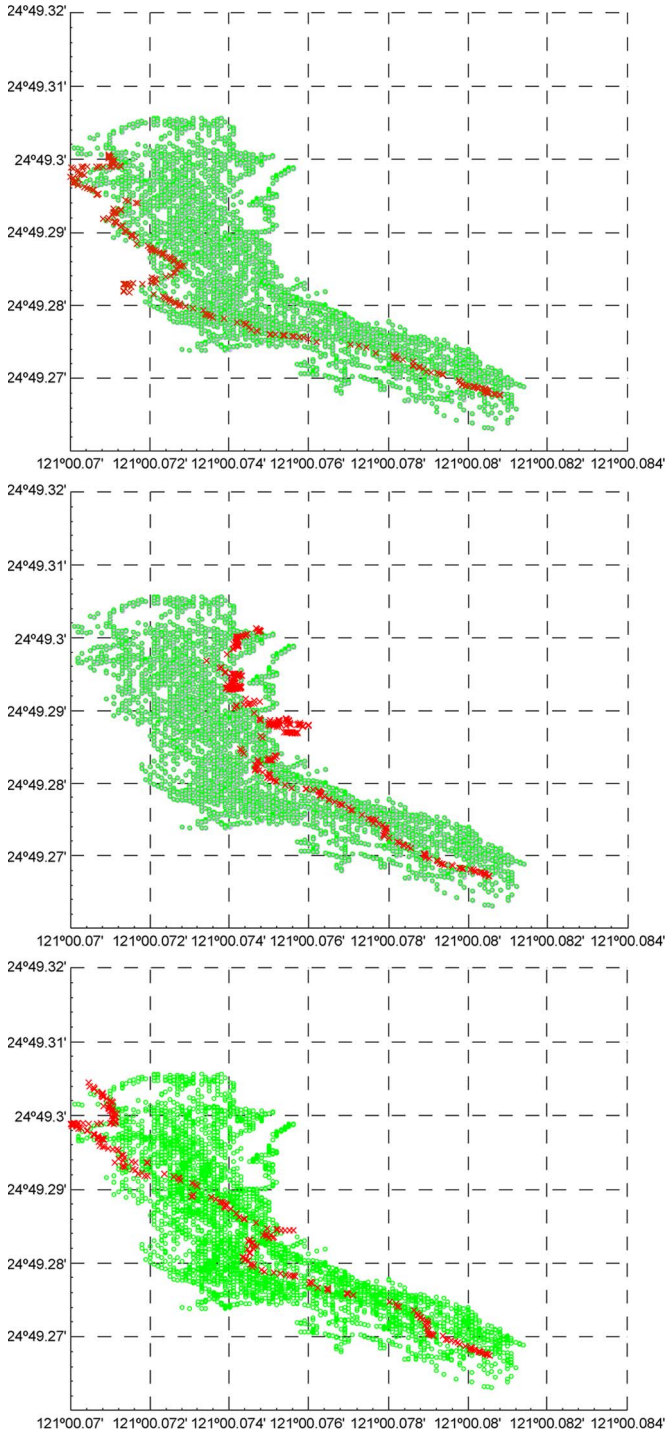


Fig. 13. Three virtual trajectories (in red) are shown over the points of the collected trajectory (in green).

statistical analyses. We generated 1229 virtual trajectories that consisted of 39×10^4 points. Their probability distributions were compared to the collected real trajectories in terms of current velocity and direction.

Figs. 14 and 15 show the velocity and direction CDFs of the collected and virtual trajectories. In Fig. 14, the velocity CDF of the virtual trajectories is compared to the CDF of the collected trajectory. In the virtual trajectories, the probability that the velocity is smaller than 2 knots is slightly higher than that in

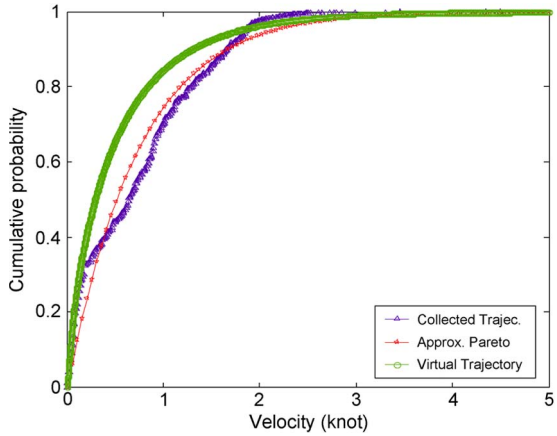


Fig. 14. Cumulative probability of current velocity.

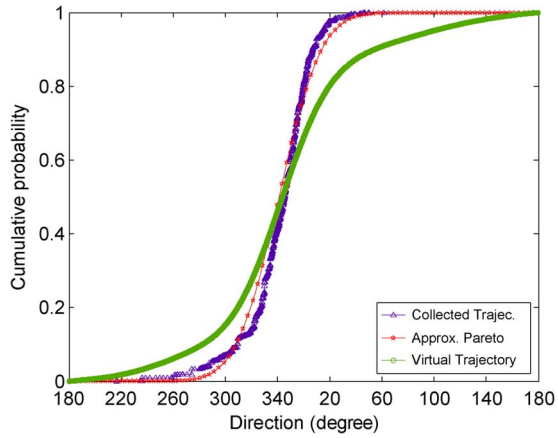


Fig. 15. Cumulative probability of current direction.

the collected trajectory. Nevertheless, the two distributions are very similar.

Fig. 15 shows the CDFs of the current direction. The direction CDF of the virtual trajectories is similar to the CDF of the collected trajectories. However, the probability curve of the virtual trajectories is more flattened than that of the collected trajectories. The reason for this might be the sample size of the virtual trajectories, which is much larger than the collected trajectory. This causes the distribution of the current direction of the virtual trajectory to approximate more toward a normal distribution.

Next, we compare the two groups of trajectories in terms of their geographical distributions. To the best of our knowledge, there is no general method that can be used to quantify the *similarity* of two sets of trajectories. Therefore, in this paper, we propose two metrics, i.e., the *coverage ratio* R_{coverage} and the *matching ratio* R_{matching}, to quantify the difference between the collected and virtual trajectories.

The *matching ratio* $R_{\text{matching}} = [0, 1]$ is adopted from pattern matching [30] in image processing, which determines the difference between two images. This is defined in this paper as

$$R_{\text{matching}} = \frac{1}{1 + \sum_{\forall C_{x,y} \in S_i} abs\left(PR(N_{x,y})\right) - PR\left(N'_{x,y}\right)}. \quad (6)$$

In the equation, $PR(N_{x,y})$ is the percentage of $N_{x,y}$ of a specific cell $C_{x,y}$ over the total number of points in the collected trajectory. $PR(N'_{x,y})$ denotes the same for the virtual trajectory. The function **abs()** returns the absolute value, while the denominator of the equation computes the summation of the differences between $PR(N_{x,y})$ and $PR(N'_{x,y})$ for all cells in S_i. A R_{matching} close to 1 indicates that the two groups of trajectories have a high similarity.

The *coverage ratio* $R_{\text{coverage}} = [0, 1]$ is defined as a relative ratio in the view of the cells covered by the collected and virtual trajectories, based on the following intuition. If two groups of trajectories (i.e., collected and virtual) are inherently identical, in any location in the target space (i.e., a cell in S_i) where the collected trajectories have visited, the generated virtual trajectories should have visited as well. R_{coverage} defines the ratio as the number of cells visited by both the collected and the virtual trajectories over the number of cells visited by the collected trajectories only. In other words, if a cell $C_{x,y}$ is *visited* by trajectory t_j, there exists a point p_k such that $p_k \in t_j$ and $p_k \in C_{x,y}$, i.e.,

$$R_{\text{coverage}} = \frac{\sum_{\forall C_{x,y} \in S_i} 1 \Big| \left(N_{x,y} > 0 \wedge N'_{x,y} > 0\right)}{\sum_{\forall C_{x,y} \in S_i} 1 \Big| \left(N_{x,y} > 0\right)}. \quad (7)$$

In (7), the nominator represents the number of cells $C_{x,y}$ in S_i in which both $N_{x,y}$ and $N'_{x,y}$ are larger than zero. That is, both the collected and virtual trajectory sets contain a least one point that has visited this cell. The denominator depicts the number of cells visited by the collected trajectories. The higher R_{coverage} is, the higher the variability in the generated trajectories becomes.

We applied the evaluation method previously described to the data collected from the proof-of-concept experiment. We also used the proposed trajectory generation algorithm to generate virtual trajectories with parameter maps from the experiment. The trajectories were generated using different levels of point quantities, ranging from 1×10^4 to 39×10^4. For each point quantity, ten sets of virtual trajectories were generated to obtain the average matching ratio and coverage ratio.

Figs. 16 and 17 show the matching ratio and coverage ratio under different point quantities. Both figures show that a larger point quantity in the virtual trajectory results in a higher matching ratio and a higher coverage ratio. In Fig. 16, there is a plateau in the matching ratio when the point quantity is 10×10^4. However, for the coverage ratio in Fig. 17, the plateau starts at 25×10^4. These results suggest that to evaluate the trajectory-generated algorithm, the point quantity of the virtual trajectory should be greater than 25×10^4 to provide a reasonable result. Fig. 17 also shows that the proposed trajectory generation function can generate highly variable trajectories, as more than 95% of the cells were visited by the virtual trajectories when the point quantity was approximately 14×10^4.

By comparing the collected and virtual trajectories using the two metrics, we can see their similarities. The virtual trajectories are able to capture the characteristics of the real environment and can thus be used to study simulations of, for example, the potential hazards caused by floating ice chunks, logs, and garbage in a waterway.

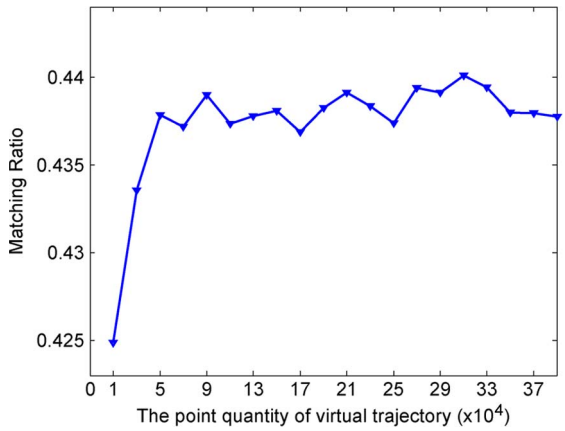


Fig. 16. *Matching ratio* versus the point quantity of the virtual trajectory.

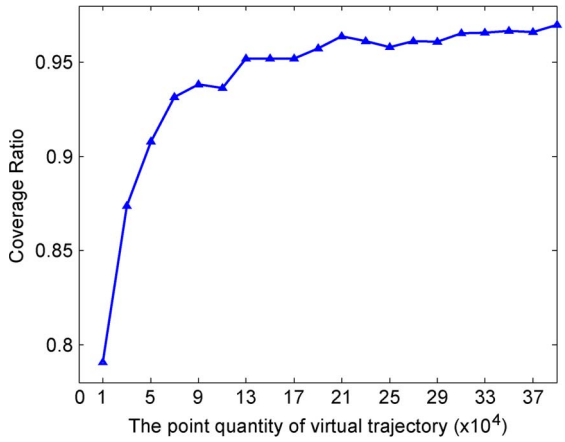


Fig. 17. *Coverage ratio* versus the point quantity of the virtual trajectory.

To summarize, in this section, we used parameter maps to build a mobility model of drifting objects and to evaluate the virtual trajectory in velocity/direction distribution, matching ratio, and coverage ratio. The results confirmed the correctness of the proposed mobility model.

V. Discussions

In this paper, we demonstrated the feasibility of the proposed method via a proof-of-concept experiment. In this section, we discuss the issues related to the accuracy of the proposed method and its uses in real-life situations.

Regarding the accuracy and potential measurement errors of the proposed method, we refer to the related works that applied a similar approach. In [33], a drifting GPS device that measures the velocity of a surface current is compared to a conventional impellor meter. The result of this paper shows that there is a strong correlation $(r > 0.8)$ between the measurements from the conventional flow meter and the GPS device. The measured average velocity error is 17.5 cm/s. Another study [34] compares the velocity measurement from a GPS on a float and a high-frequency radar system. The latter has been demonstrated to have a velocity accuracy of 10 cm/s [35], [36]. The results

from this paper show that the measured velocity difference is around 9.8–9.9 cm/s, which is close to the value shown in the datasheet of the GPS module [23] integrated in our wireless sensor platform. For safety-related applications such as those mentioned earlier in this paper, errors within centimeters may be acceptable. Again, this is application dependent.

Since the focus of this paper is on a method for measuring water currents, the accuracy and choice of the GPS module do not affect the proposed method. If a higher level of accuracy is required, we can always choose higher end GPS modules such as assisted GPS, differential GPS [37], and wide-area augmentation system [38]. Furthermore, calibration can be performed before use to control the measurement accuracy.

Our proposed method can be used as a stand-alone or to complement existing methods. For example, the sensing region of ADCP is limited, and a widespread deployment would be very expensive. We can thus deploy ADCPs at some strategic locations and use our proposed method to collect information between ADCPs. The ADCPs can further be used to calibrate the outputs from our GPS devices. This way, we can strike a balance between cost and accuracy.

The GPS sensor discussed in this paper can only float on the surface and move along with the water. However, there are applications in which the sensors have to move autonomously to certain locations or following a certain route. We may therefore enhance the sensor with an *autonomous underwater vehicle* or remotely controlled model boat (RC boat). This would also facilitate the retrieval of the sensors. If real-time information is required, we can equip the GPS devices with wireless communication capability so that they can immediately upload the collected data to the base stations on the shore. If communication range is a concern, we can drop more wireless sensors and use multihop communication to relay data. A dense deployment is feasible because our sensors are inexpensive. Even if some of these sensors do not return, they are expendable due to their low cost.

VI. Conclusion and Future Work

In this paper, we have proposed a low-cost method to measure water currents. A proof-of-concept experiment has been conducted to show the feasibility and effectiveness of this method. The obtained parameter maps provide information about the currents in a waterway and can be used to identify potential turbulence and vortexes in rivers or harbors. They can also be used to derive mobility models for drifting objects in the water. We have shown that the generated virtual trajectories are very similar to the collected trajectories and can therefore reflect real-world behaviors.

The proposed method can also be applied to other applications, such as the study of fish migration paths, renewable energy harvesting systems based on surface currents, channel safety mechanisms for boats, and how the greenhouse effect changes tides and currents offshore. We can also perform other near-surface measurements such as seawater salinity and temperature measurements by integrating appropriate sensing components into the sensing platform to provide more information for scientists to study.

Acknowledgment

The authors would like to thank P.-Z. Kuo and P.-C. Lee for their excellent technical assistance and comments. This work is based on an earlier work, "On building mobility models for floating objects," in *Proceedings of the 7th ACM Conference on Networked Embedded Sensor Systems*, Berkeley, CA, USA, pp. 377-378, 2009.

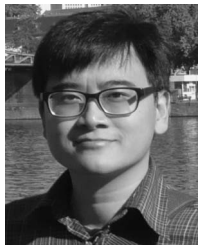
References

[1] RD Instruments, Acoustic Doppler Current Profilers-Principles of Operation: A Practical Primer, pp. 1–9, San Diego, CA: RD Instruments, 1989.
[2] G. F. Appell, P. D. Bass, and M. A. Metcalf, "Acoustic Doppler current profiler performance in near surface and bottom boundaries," *IEEE J. Ocean. Eng.*, vol. 16, no. 4, pp. 390–396, Oct. 1991.
[3] G. Appell, R. Williams, and J. Sprenke, "Development of a real time measurement system for Charleston Harbour," in *Proc. OCEANS*, Sep. 1987, vol. 19, pp. 98–103.
[4] N. Crisp and G. Griffiths, "Observations of tidal and wind-driven currents, and acoustic backscatter using an acoustic Doppler current profiler on a surface-following buoy," in *Proc. IEEE 5th Work. Conf. Current Meas.*, Feb. 7–9, 1995, pp. 233–240.
[5] W. G. Bos, "A comparison of two Doppler current profilers," *IEEE J. Ocean. Eng.*, vol. 16, no. 4, pp. 374–381, Oct. 1991.
[6] B. H. Brumley, R. G. Cabrera, K. L. Deines, and E. A. Terray, "Performance of a broad-band acoustic Doppler current profiler," *IEEE J. Ocean. Eng.*, vol. 16, no. 4, pp. 402–407, Oct. 1991.
[7] R. M. Goldstein and H. A. Zebker, "Interferometric radar measurement of ocean surface currents," *Nature*, vol. 328, no. 6132, pp. 707–709, Aug. 1987.
[8] R. Romeiser, "Current measurements by airborne along-track InSAR: Measuring technique and experimental results," *IEEE J. Ocean. Eng.*, vol. 30, no. 3, pp. 552–569, Jul. 2005.
[9] R. Romeiser, S. Suchandt, H. Runge, and U. Steinbrecher, "High-resolution current measurements from space with TerraSAR-X along-track InSAR," in *Proc. OCEANS-EUROPE*, May 11–14, 2009, pp. 1–5.
[10] H. C. Graber, D. R. Thompson, and R. E. Carande, "Ocean surface features and currents measured with synthetic aperture radar interferometry and HF radar," *J. Geophys. Res.*, vol. 101, no. C11, pp. 25 813–25 832, 1996.
[11] R. Siegmund, M. Bao, S. Lehner, and R. Mayerle, "First demonstration of surface currents imaged by hybrid along- and cross-track interferometric SAR," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 511–519, Mar. 2004.
[12] H. C. Lee, C. Y. Lin, S. W. Hsu, and C. T. King, "On building mobility models for floating objects," in *Proc. 7th ACM SenSys*, Berkeley, CA, 2009, pp. 377–378.
[13] K. Liu, M. Li, Y. Liu, M. Li, Z. Guo, and F. Hong, "Passive diagnosis for wireless sensor networks," in *Proc. 6th ACM SenSys*, Raleigh, NC, 2008, pp. 113–126.
[14] H. C. Lee, C. J. Liu, J. Yang, J. T. Huang, Y. M. Fang, B. J. Lee, and C. T. King, "Using mobile wireless sensors for in-situ tracking of debris flows," in *Proc. 6th ACM SenSys*, Raleigh, NC, 2008, pp. 407–408.
[15] H. C. Lee, A. Banerjee, Y. M. Fang, B. J. Lee, and C. T. King, "Design of a multi-functional wireless sensor for in-situ monitoring of debris flows," *IEEE Trans. Instrum. Meas.*, to be published.
[16] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. MOBICOM*, Dallas, TX, 1998, pp. 85–97.
[17] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proc. MOBICOM*, Seattle, WA, 1999, pp. 195–206.
[18] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proc. ACM MSWiM*, Seattle, WA, 1999, pp. 53–60.
[19] C. Tuduce and T. Gross, "A mobility model based on WLAN traces and its validation," in *Proc. IEEE INFOCOM*, Miami, FL, 2005, pp. 664–674.
[20] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces," in *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006, pp. 1–13.
[21] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. 4th IPSN*, Los Angeles, CA, 2005, pp. 364–369.
[22] MSP430. [Online]. Available: http://focus.ti.com/mcu/docs/mcuprodoverview.tsp?sectionId=95&tabId=140&familyId=342
[23] FPGMMOPA1. [Online]. Available: http://www.f-tech.com.tw

[24] [Online]. Available: http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp
[25] [Online]. Available: http://www.mathworks.com/
[26] J.-Y. Le Boudec and M. Vojnovic, "Perfect simulation and stationarity of a class of mobility models," in *Proc. IEEE INFOCOM*, Miami, FL, 2005, pp. 2743–2754.
[27] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Proc. ACM MOBICOM*, San Diego, CA, 2003, pp. 217–229.
[28] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput. (WCMC)—Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, Aug. 2002.
[29] W. J. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proc. 26th IEEE INFOCOM*, May 2007, pp. 758–766.
[30] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. London, U.K.: Chapman & Hall, 1998.
[31] M. Zhao and W. Wang, "Design and applications of a smooth mobility model for mobile ad hoc networks," in *Proc. IEEE MILCOM*, Oct. 23–25, 2006, pp. 1–7.
[32] J. Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for mobile ad hoc network simulation," in *Proc. 35th Annu. Simul. Symp.*, Apr. 14–18, 2002, pp. 337–344.
[33] R. J. Stockdale, S. J. McLelland, R. Middleton, and T. J. Coulthard, "Measuring river velocities using GPS river flow tracers (GRiFTers)," *Earth Surf. Process. Landforms*, vol. 33, no. 8, pp. 1315–1322, Jul. 2008.
[34] J. C. Ohlmann, P. F. White, A. L. Sybrandy, and P. P. Niiler, "GPS cellular drifter technology for coastal ocean observing systems," *J. Atmos. Ocean. Technol.*, vol. 22, no. 9, pp. 1381–1388, Sep. 2005.
[35] J. D. Paduan and L. K. Rosenfeld, "Remotely sensed surface currents in monterey bay from shore-based HF radar (coastal ocean dynamics application radar)," *J. Geophys. Res.*, vol. 101, no. C9, pp. 20 669–20 686, 1996.
[36] R. D. Chapman and H. C. Graber, "Validation of HF radar measurements," *Oceanography*, vol. 10, pp. 76–79, 1997.
[37] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal, "GPS: Location-tracking technology," *Computer*, vol. 35, no. 4, pp. 92–94, Apr. 2002.
[38] P. Enge, T. Walter, S. Pullen, C. Kee, Y. C. Chao, and Y. J. Tsai, "Wide area augmentation of the Global Positioning System," *Proc. IEEE*, vol. 84, no. 8, pp. 1063–1088, Aug. 1996.

Huang-Chen Lee (S'09) received the M.S. degree in computer science from Soochow University, Taipei, Taiwan, in 2005. He is currently working toward the Ph.D. degree in computer science with the National Tsing Hua University, Hsinchu, Taiwan.

He has worked in the industry since 2000 and has a wide breadth of experience in designing PDA/cellular phones and low power embedded systems. His research interests include distributed processing and networked embedded systems.



Chun-Yu Lin received the B.S. degree in computer science from Soochow University, Taipei, Taiwan, in 1999 and the M.S. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 2001,where he is currently working toward the Ph.D. degree.

His research interests are in the areas of wireless sensor networks, pervasive computing, and data mining.

Chun-Han Lin (S'09) received the B.S. and M.S. degrees in computer science from the National Chiao Tung University, Hsinchu, Taiwan, in 1999 and 2001, respectively. He is currently working toward the Ph.D. degree with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan.

His research interests are in the areas of wireless sensor network, optimization problem, and power conservation technology.

Sheng-Wen Hsu received the M.S. degree in computer science from the Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan, in 2008.

From 2008 to 2009, she was a Wi-Fi Engineer with the Ralink Technology Corporation. In 2009, she joined MediaTek, where she is currently an Engineer focused on GPS receiver. She is expert at WLAN protocol stack, GPS receiver system, RTOS, and embedded system.

Chung-Ta King (M'88–SM'06) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1980 and the M.S. and Ph.D. degrees in computer science from Michigan State University, East Lansing, in 1985 and 1988, respectively.

From 1988 to 1990, he was an Assistant Professor of computer and information science with the New Jersey Institute of Technology, Newark. In 1990, he joined the faculty of the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, where he is currently a Professor and the Department Chair. His research interests include parallel and distributed processing and networked embedded systems.