Track Estimation Using Link Line Crossing Information in Wireless Networks

Peter Hillyard[1,3], Samira Daruki[2], Neal Patwari[1,3], and Suresh Venkatasubramanian[2]

[1]Dept. of Electrical & Computer Engineering, University of Utah, Salt Lake City, USA,
[hillyard,npatwari]@ece.utah.edu
[2]School of Computing, University of Utah, Salt Lake City, USA, [daruki,suresh]@cs.utah.edu
[3]Xandem Technology, Salt Lake City, USA

Abstract—**Device-free or non-cooperative localization uses the changes in signal strength measured on links in a wireless network to estimate a person's position in the network area. Existing methods provide an instantaneous coordinate estimate via radio tomographic imaging or location fingerprinting. In this paper, we explore the problem of, after a person has exited the area of the network, how can we estimate their path through the area? We present two methods which use recent line crossings detected by the network's links to estimate the person's path through the area. We assume that the person took a linear path and estimate the path's parameters. One method formulates path estimation as a line stabbing problem, and another method is a linear regression formulation. Through simulation we show that the line stabbing approach is more robust to false detections, but in the absence of false detections, the linear regression method provides superior performance.**

I. Introduction

Current research in device-free localization can provide instantaneous or real-time coordinate estimates of a person inside the deployment area of a wireless network. The person carries no wireless device to assist in the localization. Technologies such as radio tomographic imaging (RTI) can achieve remarkable localization accuracy [3], [6]. However, the accuracy comes at the requirement of a high density of radio sensing nodes.

In this paper, we explore the estimation of the person's path or trajectory through the deployment area after the person has travelled through. We may want to know the direction and velocity of their path, and be willing to wait until the person has crossed through in order to make the estimate. In cases when the node (and thus link) density is low, a person may cross links only occasionally, and thus instantaneous point estimates may have significantly higher error than the after-the-fact path estimate.

We propose two path estimators which receive which link lines were detected as crossed and at what times, and output the estimates of the parameters of a linear path. We approximate the person's path as straight based on the observation that people tend to walk in piecewise straight paths. The first method formulates path estimation as a line stabbing problem, and the second method is a linear regression formulation.

Inaccuracies that arise as a result of detecting link line crossings are the major hurdles affecting the performance of our proposed methods. Link line crossing detection methods will ultimately miss crossing events and will report false crossings. The solutions we propose track a person's path in spite of these errors. We analyze the performance of the two methods through simulation.

The paper is organized as follows. We introduce our link line detection model, and the line stabbing and linear regression algorithms in Section II. We then present a simulation framework and provide and discuss results in Section III. We conclude in Section IV.

II. Methodology

In this paper, we consider a wireless network with N wireless devices, or *nodes*. The coordinate of the ith node is given as $\mathbf{z_i} = (x_i, y_i)$ in a Cartesian plane. In this work we consider only 2D localization. The link $k = (i, j)$ is a communication link between transmitter i and receiver j. We refer to the *link line segment* as s_k, whose endpoints are $\mathbf{z_i}$ and $\mathbf{z_j}$; and to the *link line* which contains both $\mathbf{z_i}$ and $\mathbf{z_j}$. Link line k is also given as the equation $a_k y = b_k x + c_k$. In this paper, we assume a single person is walking, with position $\mathbf{p}[n]$ at time index n. Time index n corresponds to time nT_s, where T_s is a sampling period.

We assume that link line crossing detection is performed using measurements of the radio channel. For example, the channel measurement might be the RSS. When a person passes near a link line segment, he tends to affect the received signal strength (RSS) measured on the link, in a manner that is detectable [2], [5], [6]. We assume a detector as in [2], [5], [6] is applied to the channel measurements at each time index.

A. Link Line Crossing Detection Model

Link crossing detection methods are prone to both false alarms and missed detections [6], but our experimental experience tells us that the further a person is from a link line segment, the less likely that link will detect a crossing [4]. Currently, no statistical model for link line crossing detection performance has been reported in the literature. In this work, we use a simple model to describe the probability of a link's channel measurement being "perturbed" by a person's presence at a particular location, and that perturbation then causing the link to be detected as "crossed". We denote $P_k[n]$

as the probability that link $k = (i, j)$ is detected as *perturbed* at time index n,

$$P_k[n] = P_0 e^{-\frac{1}{\delta}(\|\mathbf{z}_i - \mathbf{p}[n]\| + \|\mathbf{z}_j - \mathbf{p}[n]\| - \|\mathbf{z}_i - \mathbf{z}_j\|)} \quad (1)$$

where P_0 is the probability a perturbation event is detected when a person is blocking the link line segment, and δ is the decay constant. A smaller δ value models links that experience changes in RSS solely when $\mathbf{p}[n]$ is very near the link, while larger δ values model links that experience significant changes in RSS even when $\mathbf{p}[n]$ is relatively far away.

Link line crossing algorithms consider a link line *crossed* only when detected as perturbed consecutively for longer than T_c seconds, or $n_c = \lfloor \frac{T_c}{T_s} \rfloor$ samples [5]. If the case arises that a link detects more than one perturbation, the system detects a crossing and records the median of the perturbation events as the crossing time. After the person crosses through the wireless network, the system generates a vector $Q_{state} = [q_1, q_2, \dots, q_L]^T$ where L is the number of links and q_k is 0 when link k is not detected as crossed and 1 when it is detected as crossed. The system also generates $T_{time} = [t_1, t_2, \dots, t_L]^T$ where t_k is the time in seconds when link k is detected as crossed. We use Q_{state} and T_{time} in the linear regression and line stabbing algorithms to estimate the person's path as described in the following subsections.

B. Line Stabbing Estimation

In computational geometry, given a set of line segments, the line stabbing problem is to define the set of lines $y = mx + b$ that intersect (or *stab*) all the line segments [1]. In our case, the line segments are the link line segments $\{s_k\}_k$, and we wish to estimate a line which corresponds to the straight path of a person passing through a wireless network. The key challenge in our problem is that the system may record false link crossings or miss link crossings.

To solve this problem, we use the principle of point-line duality. A point with coordinates (a, b) is transformed to a line $l^* = ax + (-b)$ and a line $l' = ax + b$ is transformed to a point with coordinates $(a, -b)$. A line segment, which is defined by two endpoints p_1 and p_2 is transformed into a dual-wedge, where the boundaries of the dual-wedge are the lines l_1^* and l_2^* and the region between the boundaries are the infinite number of lines that pass through the intersection of l_1^* and l_2^*. Overlapping dual-wedges form a *stabbing region* such that any point p_{stab} within the region transforms into a line l_{stab} that stabs the set of line segments that correspond to the overlapping dual-wedges.

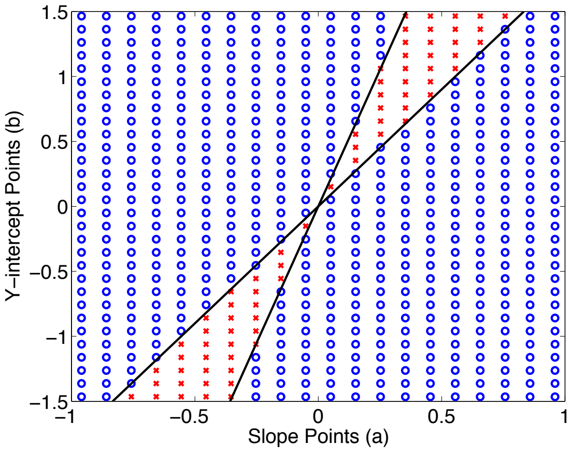

Fig. 1: An example matrix M_k containing the dual-wedge w_k. Blue o's are where $M_k(a, b) = 0$ and red x's are where $M_k(a, b) = 1$. The black lines are the dual of the endpoints of line segment s_k.

In this work, the L links in the network are represented as line segments s_k, where k corresponds to the kth link. The line segment s_k is transformed into its dual-wedge w_k. For speed of processing, we use a discretized grid of points, denoted by the matrix M_k, to define w_k: the grid point $M_k(a, b)$ is assigned the value 1 if (a, b) is inside w_k and 0 if (a, b) is outside w_k. The grid points for all M_k are bounded horizontally by $[a_{min}, a_{max}]$ and vertically by $[b_{min}, b_{max}]$. An example of a discrete dual-wedge is shown in Fig. 1. We combine all L matrices in a database C, where the 1^{st} and 2^{nd} dimensions are the grid points and the 3rd dimension is the kth matrix, M_k. If we were to examine $C(0.3, 0.6)$, we would see a $1 \times L$ binary vector, $C(0.3, 0.6) = [c_1, c_2, \dots, c_L]$. If $c_k = 1$, then the point $(0.3, 0.6)$ is inside w_k, and is outside w_k if $c_k = 0$. Applying duality to point $(0.3, 0.6)$, the line $l^* = 0.3x + (-0.6)$ stabs all line segments s_k whose corresponding c_k value is 1. Suppose that the link crossing detection system could perfectly detect link crossings. Then we could define a set of lines that stab the crossed links by finding all points in $C(a, b)$ where $Q_{state} = C(a, b)$. But since the link crossing detection system misses and falsely detects link line crossings, we are left to find all points in $C(a, b)$ such that the difference in values in $C(a, b)$ and Q_{state} are minimized. We define the stabbing region to be the set of points (a, b) such that the Hamming distance between $C(a, b)$ and Q_{state} is minimized.

$$\underset{(a,b)}{\arg\min}\{d(Q_{state}, C(a, b))\} \quad (2)$$

where $d(Q_{state}, C(a, b)) = |\{i : q_i - c_i \neq 0\}|$, *i.e.*, the number of disagreements between the two vectors. In many cases, performing (2) yields multiple points (a, b) that define the stabbing region. We require one point, p_{stab}, whose dual is an estimate of the path traveled by the person. In this paper, we choose the centroid of points produced by (2) to be p_{stab}, whose dual is l_{est}. Estimating p_{stab} with a maximum liklihood or Bayesian estimator, is left to future work.

C. Linear Regression Estimation

In addition to the line stabbing adaptation, we also use ordinary least squares to estimate the initial position and trajectory of a person walking through the network using link crossing information in Q_{state} and the detected time of crossing in T_{time}. The path the person travels is parameterized such that when he crosses link k at time t_k, his coordinate (x, y) is defined by

$$\begin{aligned} x &= t_k v_x + p_x \\ y &= t_k v_y + p_y \end{aligned} \quad (3)$$

where the unknown parameters $\mathbf{p}[0] = [p_x, p_y]^T$ are the coordinates of the person at $t = 0$ and $[v_x, v_y]^T$ are the rate of change of the x and y coordinates, respectively. For all links k for which $q_k = 1$, we combine the line equation of all links detected as crossed, their recorded crossing time, and (3) to form a matrix equation

$$\underbrace{\begin{bmatrix} a_1t_1 & -b_1t_1 & a_1 & -b_1 \\ a_2t_2 & -b_2t_2 & a_2 & -b_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_Rt_R & -b_Rt_R & a_R & -b_R \end{bmatrix}}_{\alpha} \underbrace{\begin{bmatrix} v_y \\ v_x \\ p_y \\ p_x \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_R \end{bmatrix}}_{\beta}. \quad (4)$$

The value R must be ≥ 4 and is the number of links detected as crossed. Given that the link crossing detection and timing system are imperfect, we minimize the error in θ by using an ordinary least squares estimation:

$$\hat{\theta} = (\alpha^T\alpha)^{-1}(\alpha^T\beta). \quad (5)$$

The line that estimates the path the person took in the network is defined as $l_{est} = (\hat{v}_y/\hat{v}_x)x - (\hat{v}_y/\hat{v}_x)\hat{p}_x + \hat{p}_y$.

D. Quantifying Estimation Accuracy

In this section, we describe how we quantify the accuracy of the line estimate l_{est} obtained from the algorithm in Section II-B or II-C. While many quantification methods exist, we calculate accuracy as the root-mean-square error (RMSE) of a discretized line segment that passes through the area. We calculate the bottom-left-most and top-right-most intersection points between that of the true path and the network as well as the estimated path and the network. These coordinates are denoted as $\mathbf{z}_{ll}^{path}$, $\mathbf{z}_{ur}^{path}$, and $\mathbf{z}_{ll}^{est}$, $\mathbf{z}_{ur}^{est}$. We choose P evenly distributed points along the lines connecting the coordinates of $\mathbf{z}^{path}$ and $\mathbf{z}^{est}$ and save both set of points in the $P \times 2$ matrices D_{path} and D_{est}. We calculate the RMSE as,

$$\text{RMSE} = \sqrt{\frac{1}{P}\|D_{path} - D_{est}\|_F}, \quad (6)$$

where $\|\cdot\|_F$ indicates the Frobenius norm.

III. Results

In this section, we describe our simulation framework, and then use it to show the accuracy of the line stabbing and linear regression methods.

A. Simulation Framework

In simulation, 30 nodes are placed inside of an 276 m^2 area as shown in Fig. 2. A link $k = (i, j)$ is formed between the nodes only when $\|\mathbf{z_i} - \mathbf{z_j}\| < d_{max}$, where d_{max} is the connectivity distance. We use $d_{max} = 10$m in this simulation. The endpoints of the true path for each simulation are normally distributed $N(0, 1)$ around the points $(0, -5)$ and $(30, 10)$. We also adjust P_0 and δ in (1) for each trial to simulate scenarios where the detection rate (*i.e.*, the ratio of accurately detected link line crossings to the total number of true link line crossings) and false alarm rate (*i.e.*, the ratio of the number of false alarms to the product of the number of samples taken by the wireless network and the number of links) are high and low (see Figures 3 and 4). For each trial, we record the false alarm rate, the detection rate, and the RMSE with the associated value of P_0 and δ. In this work, anywhere from sixty to seventy link lines are crossed for each trial.

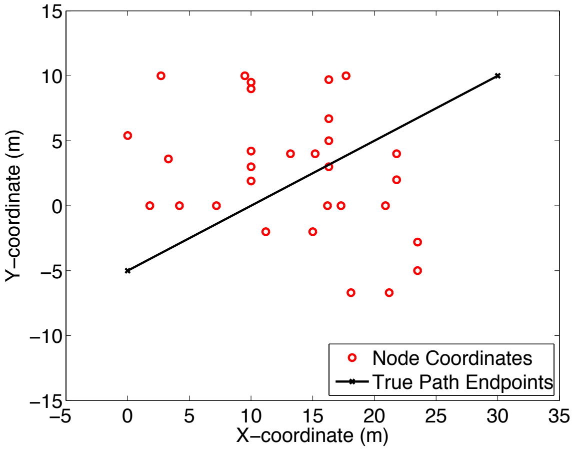

Fig. 2: Node coordinates in the simulation network. Endpoints of the true path are normally distributed around the lower and upper (x).

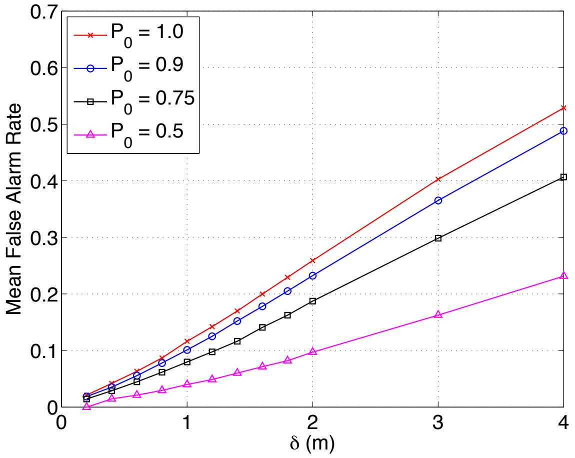

Fig. 3: Mean false alarm rate as a function of P_0 and δ.

B. Line Stabbing Performance

Fig. 5 shows the relationship between P_0 and δ and $RMSE_{avg}$. For $P_0 \geq 0.75$, $RMSE_{avg}$ is ≈0.8 m for small δ and increases to >2.0 m for large δ. We note the unexpected influence of the parameter P_0 — for almost all values of δ, the lower the value of P_0, the lower the $RMSE_{avg}$. This unexpected behavior is due to the false alarm rate vs. P_0 and δ, as seen in Fig. 3. Lower values of P_0 result in lower false alarm rates over all δ. Although the detection rate is higher for larger P_0 (see Fig. 4), $RMSE_{avg}$ increases for larger P_0 because of the higher false alarm rate.

C. Regression Performance

In Fig. 6, we show the $RMSE_{avg}$ of all simulations with the given P_0 and δ value. We observe that the $RMSE_{avg}$ has

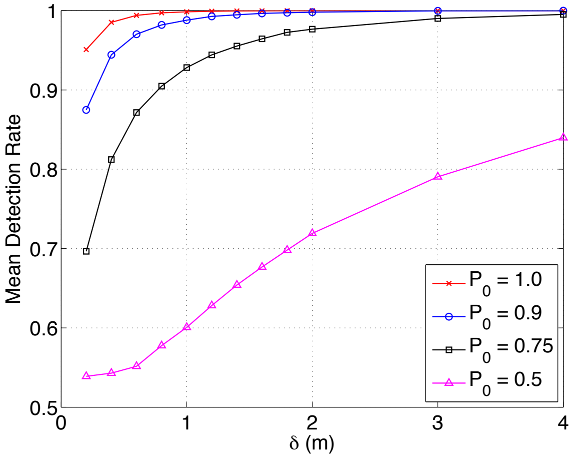

Fig. 4: Mean detection rate as a function of P_0 and δ.

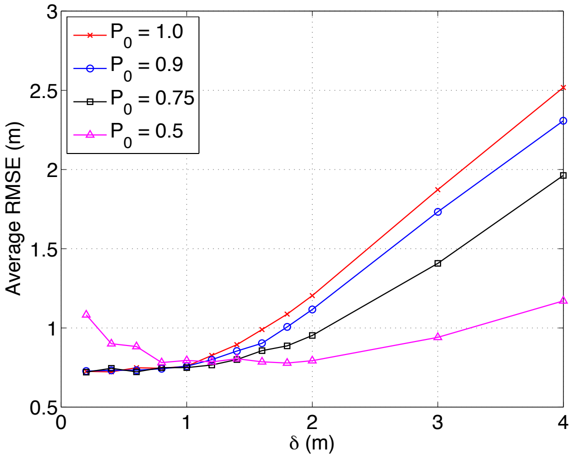

Fig. 5: Average RMSE for the line stabbing algorithm as a function of P_0 and δ.

a minimum of ≈0.5 meters when δ is small for all values of P_0. As δ increases, $RMSE_{avg}$ also increases until it peaks at ≈2.5 meters when $\delta = 2$ meters for $P_0 \geq 0.75$. This trend can be attributed to the average false alarm rate at each value of δ, as observed in Fig. 3. For all values of P_0, the false alarm rate increases as δ increases. A higher false alarm rate tends to increase $RMSE_{avg}$ until it reaches a maximum. It should also be noted that although the system achieves a high detection rate for large δ (see Fig. 4), the $RMSE_{avg}$ does not reduce significantly.

D. Performance Comparison

We draw a few conclusions about the different performance of the line stabbing and linear regression algorithms. Using the linear regression algorithm, $RMSE_{avg}$ maxes out at ≈2.5 meters whereas the $RMSE_{avg}$ in the adapted line stabbing algorithm would eventually exceed 2.5 meters for $\delta > 4$ meters (as observed in Figures 5 and 6). However, if $0.1 < \delta < 4$, the adapted line stabbing algorithm will provide a more accurate best line estimate (regardless of the the detection rate). In other words, the line stabbing approach is more robust to false alarms; but in the absence of false alarms, the linear regression algorithm provides the better accuracy. Note that low δ implies that links are only perturbed when a person is very close to the line between the transmitter and receiver.

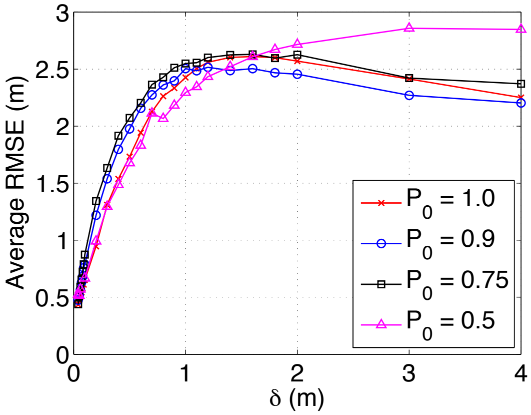

Fig. 6: Average RMSE for the regression algorithm as a function of P_0 and δ.

IV. Conclusion

In this paper, we present algorithms for device-free tracking using a wireless network. We study the algorithms via simulation using a simple probability of link line crossing detection model. Simulations show it is possible to track the straight path of a person passing through a wireless network despite the presence of false and missed line crossing detections. When there are few false alarms, we can accurately estimate the track of the person to within a RMSE of 0.8 meters using the adapted line stabbing algorithm and 0.5 meters using the linear regression algorithm. However, the line stabbing algorithm is more robust to a high false alarm rate compared to the linear regression approach.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. 0748206 and 1035565.

References

[1] H. Edelsbrunner, H. Maurer, F. Preparata, A. Rosenberg, E. Welzl, and D. Wood. Stabbing line segments. *BIT Numerical Mathematics*, 22(3):274–281, 1982.

[2] O. Kaltiokallio and M. Bocca. Real-time intrusion detection and tracking in indoor environment through distributed rssi processing. In *2011 IEEE 17th Intl. Conf. Embedded and Real-Time Computing Systems and Applications (RTCSA)*, volume 1, pages 61 –70, Aug. 2011.

[3] O. Kaltiokallio, M. Bocca, and N. Patwari. Follow @grandma: Long-term device-free localization for residential monitoring. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 991–998, 2012.

[4] N. Patwari and J. Wilson. Spatial models for human motion-induced signal strength variance on static links. *Information Forensics and Security, IEEE Transactions on*, 6(3):791–802, 2011.

[5] M. Youssef, M. Mah, and A. Agrawala. Challenges: device-free passive localization for wireless environments. In *MobiCom '07: ACM Int'l Conf. Mobile Computing and Networking*, pages 222–229, 2007.

[6] Y. Zhao, N. Patwari, J. M. Phillips, and S. Venkatasubramanian. Radio tomographic imaging and tracking of stationary and moving people via kernel distance. In *Proceedings of the 12th international conference on Information processing in sensor networks*, IPSN '13, pages 229–240, New York, NY, USA, 2013. ACM.