A Wearable Interface for Topological Mapping and Localization in Indoor Environments

Grant Schindler[1], Christian Metzger[2], and Thad Starner[1]

[1] Georgia Institute of Technology, College of Computing, Atlanta, GA, USA
{schindler, thad}@cc.gatech.edu
[2] ETH-Swiss Federal Institute of Technology, Information Management, Zurich, Switzerland
cmetzger@ethz.ch

Abstract. We present a novel method for mapping and localization in indoor environments using a wearable gesture interface. The ear-mounted FreeDigiter device consists of an infrared proximity sensor and a dual axis accelerometer. A user builds a topological map of a new environment by walking through the environment wearing our device. The accelerometer is used to identify footsteps while the proximity sensor detects doorways. While mapping an environment, finger gestures are used to label detected doorways. Once a map is constructed, a particle filter is employed to track a user walking through the mapped environment while wearing the device. In this tracking mode, the device can be used as a context-aware gesture interface by responding to finger gestures differently according to which room the user occupies. We present experimental results for both mapping and localization in a home environment.

1 Introduction

We are interested in quickly and automatically mapping indoor environments to facilitate context-aware wearable computing interfaces. When brought into an unknown environment, an ideal wearable device would map the environment with little user intervention and then continue to track the user's movements throughout the constructed map. Though the mobile robotics community has studied this same problem for decades, many of their solutions involve sensors (e.g. laser range finders) which are either too large, too heavy, or too expensive to be built into a wearable computer. Instead, for this task we propose a cheap, lightweight device requiring minimal user intervention.

The device we use for this task, the FreeDigiter, was conceived as a contact-free finger-gesture interface for mobile devices [7]. It is equipped with a dual axis accelerometer and uses an infrared proximity sensor to detect numerical finger gestures. We extend the capabilities of this interface to detect when the user has walked through a doorway, thus enabling the construction of topological maps of new environments and the subsequent localization of users within these environments.

The contribution of our approach is in solving the mapping and localization problems for the case of extremely sparse sensor data from inexpensive wearable hardware – a situation which necessitates the development of a unique map representation to make these tasks possible. The implications of a cheap, wearable mapping and

M. Hazas, J. Krumm, and T. Strang (Eds.): LoCA 2006, LNCS 3987, pp. 64–73, 2006.

Fig. 1. The FreeDigiter device is equipped with a dual axis accelerometer, an infrared proximity sensor, and a Bluetooth radio for wireless communication. The entire device is worn like a pair of headphones with the sensors positioned over one of the ears.

localization device go beyond domestic applications, extending to fire, rescue, and other dangerous situations involving unknown indoor environments.

2 Related Work

In their location-recognition work, Lee and Mase [5, 4] have used wearable accelerometers and other sensors to recognize when users perform specific activities and transition between known indoor locations. They use a dead reckoning approach, integrating accelerometer data over time to build a metric map of a user's path through an environment. While we similarly count a user's steps, we take an otherwise different approach by building and tracking in topological maps which capture the connectivity of an indoor environment composed of multiple rooms.

Much work has also been done in the field of robotics on the dual problems of mapping and localization. Simmons and Koenig did work on mobile robot navigation in partially observable environments in which they integrated topological maps with metric information for localization purposes [10]. The augmented topological maps they use are constructed by hand and the intended movements of the robot are known. In contrast, we must infer the movements of the user through the environment from sensor information and automatically construct maps of the environment.

A large portion of the robotics literature focuses on Simultaneous Localization And Mapping (SLAM). For example, Howie Choset discusses the use of the Generalized Voronoi Graph [2, 3], a specific topological map, for mapping and navigating in unknown environments. This work has recently been applied to the problem of person tracking [6] by placing sensors throughout a known environment and employing a particle filter to track multiple individuals. For tracking, we also take an approach based on particle filtering, but we have only one sensor module, our sensors are mobile, and we know nothing about our environment.

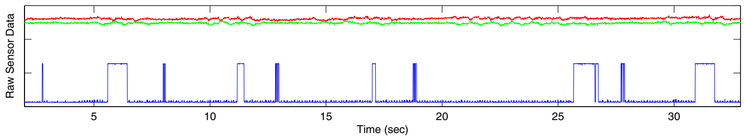


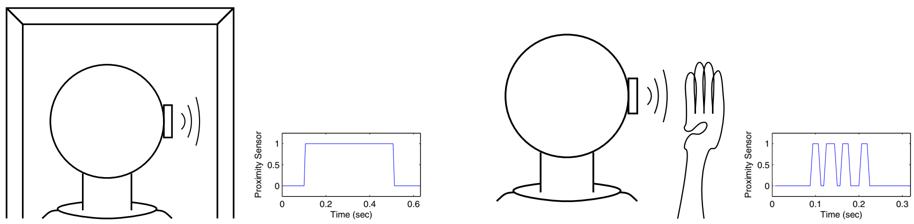
Fig. 2. Raw sensor data. The three data streams include dual axis accelerometer data (top) and proximity sensor data (bottom). The top (red) signal indicates side-to-side motion, the middle (green) signal shows front-to-back motion, and spikes in the bottom (blue) signal indicate finger or door detection events by the proximity sensor. All mapping and localization is performed utilizing only these sensor readings.

3 Sensors

The FreeDigiter is equipped with a dual axis accelerometer and an infrared proximity sensor. By processing the signals returned by each of these sensors (see Figure 2), we are able to detect footsteps, doorways, and finger gestures.

3.1 Proximity Sensor

The proximity sensor emits infrared light and detects reflected light from objects within a range of 10cm to 60cm. The output from the sensor is binary, indicating whether there is or is not an object within range. Originally intended as a method for input of numerical finger gestures, the FreeDigiter counts the number of peaks in the proximity detector signal as a user sweeps some number of fingers past the sensor. We extend the functionality of the proximity sensor by also detecting when a wearer has passed through a doorway. To the sensor, the difference between a finger and a doorway is the length of the pulse generated when each object passes (see Figure 3). Due to the high temporal resolution of the proximity sensor (160 Hz), even differently-shaped doorways will generate unique pulse-lengths.



(a) Doorway (b) Fingers

Fig. 3. The proximity sensor worn over the ear detects both a) when the user passes through a narrow space such as a doorway and b) when the user waves any number of fingers past the sensor. The two cases are reliably distinguished by the durations of spikes in the respective signals.

3.2 Accelerometer

We are particularly interested in detecting when a person wearing our device has taken a step forward. There is a large body of work in wearable computing that makes use of accelerometers for motion analysis[1, 8]. In many cases, accelerometers are placed on the feet or legs to extract walking data. Our device is head-mounted, and we found the most dependable footstep information comes from the x-axis accelerometer measuring motion perpendicular to the direction of walking (i.e. left-to-right motion). Though the raw data is rather noisy, it has a consistent sinusoidal motion during walking that can be examined with a low-pass filter (see Figure 4). Each peak and trough of this signal is a step with the left and right foot, respectively. Thus, we can define a footstep as a zero-crossing of the first derivative of the low-pass-filtered accelerometer data. The reliability of footstep detection provided by this method is suitable for our purposes.

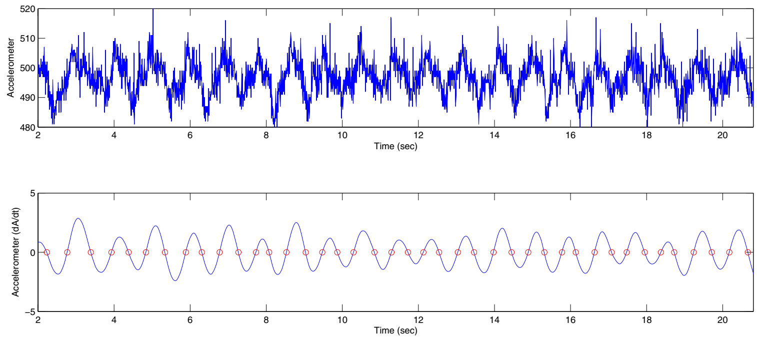


Fig. 4. The raw accelerometer data (top) for left-right motion is quite noisy. We can accurately and reliably detect footsteps (red circles in the bottom figure) as zero-crossings of the first derivative of the signal convolved with a Gaussian.

4 Mapping and Localization

We are interested in not only mapping an unknown indoor environment, but also in using the constructed map to track a user moving through the rooms of that environment. Thus, we need a representation that is easy to construct from our limited stream of data and that also has enough descriptive power to allow us to discriminate between features of the environment. For this purpose, we make use of an augmented topological map, similar to the maps used by Simmons and Koenig [10].

A map is represented as a set of edges E and vertices V defining a graph $G = \{E, V\}$. Each edge is augmented with a length l (in footsteps) and an edge-specific probability distribution over proximity sensor readings (the mean μ and variance σ of a Gaussian). By this definition, each edge corresponds to a constant part of the environment – i.e. the world looks the same to the sensors at every point on an edge. We define two types of edges:

1. **undirected edges** - Any edge $E_{i,j}$ between two different doorways with vertices V_i and V_j represents an undirected path across a room, where $l_{E_{i,j}}$ is the length of the

edge in footsteps. For undirected edges, we specify a probability distribution over proximity sensor readings with $\{\mu_{E_{i,j}} = 0, \sigma_{E_{i,j}} = 1\}$ since no doorways will be detected while traversing undirected edges inside of a room.

2. **directed edges** - We represent every doorway as a pair of vertices V_k and V_{-k} connected by two oppositely directed edges (an entrance $E_{k,-k}$ and an exit $E_{-k,k}$), each having its own distribution over sensor readings parameterized by $\{\mu_{E_{k,-k}}, \sigma_{E_{k,-k}}\}$ and $\{\mu_{E_{-k,k}}, \sigma_{E_{-k,k}}\}$. By convention, doorway edges are one footstep long ($l_{E_{k,-k}} = l_{E_{-k,k}} = 1$).

Given this representation, it follows that any clique of nodes in the graph G connected solely by directed edges represents a doorway, while a room is defined as a clique of nodes connected by undirected edges in the graph. Rooms that are "dead ends" consist of a single edge looping back to the same vertex.

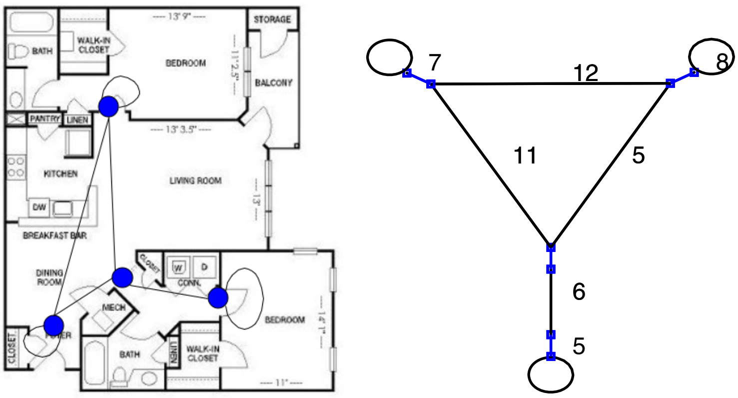


Fig. 5. Left: Topological map constructed by hand and overlaid on a metric map of our indoor environment. Right: Augmented topological map constructed automatically by walking through the physical environment. Each undirected black edge is labeled with edge length in footsteps. Each blue edge is really a pair of oppositely directed edges that correspond to the two ways of passing through a doorway, each of which is associated with its own probability distribution over proximity sensor readings.

4.1 Mapping

To map an unknown environment, a user wearing our device walks through the environment, passing through doorways and moving between rooms that should be connected in the graph. In building these maps, we will rely on a proximity sensor to detect and measure doorways and use an accelerometer to determine the distance between doorways, as outlined in Section 3, but this is not enough. When building a topological map, there is ambiguity in knowing when one has come back to the same place – e.g. traveling in a loop, or simply exiting through a door one had entered through previously in the opposite direction. Also note that the proximity sensor only sits on one side of the head. Thus, it will not have seen both sides of the doorway on the way into a room, and

Algorithm 1. Building an Augmented Topological Map

$E = \{\}\,,\, V = \{\}$ //empty graph
$while\ (BuildingMap)$
  $if\ (ProximitySensor.DoorDetected \quad \&\&$
    $ProximitySensor.FingerGesture)$ //detected and labeled door
    $k = ProximitySensor.Label$ //numerical label from finger gesture
    $if\ (V_k \cap V = \{\})$ //new doorway
      $V = V \cup \{V_k, V_{-k}\}$ //create vertices for new doorway
      $new = k$
    $else\,if\ (V_{-k} \cap V = \{\})$ //known doorway from different direction
      $new = -k$
    $end\,if$
    $E = E \cup \{E_{new,-new}\}$ //add directed edge through doorway
    $\mu_{E_{new,-new}} = ProximitySensor.Measurement$ //augment edge with sensor reading
    $l_{E_{new,-new}} = 1$ //augment edge with footstep length
    $if\ (E = \{\})\ old = new$ //first doorway
    $E = E \cup \{E_{old,new}\}$ //add undirected edge
    $l_{E_{old,new}} = Accelerometer.Footsteps$ //augment edge with footstep length
    $Accelerometer.Footsteps = 0$
    $old = new$
  $end\,if$
  $if\ (Accelerometer.Step)\ Accelerometer.Footsteps++$
$end\,while$

the two sides of a doorway can look vastly different to the proximity sensor. Though there are methods for inferring topology from unlabeled landmarks [9], we choose to avoid this problem by letting the user indicate, through numerical finger gestures, labels for each doorway after s/he passes through it. Thus, if a user waves four fingers past the sensor after entering through the front door of a building, then when the user later exits through that front door, s/he should also signal with four fingers, providing a unique identifier for each doorway. Note that this user interaction is only needed during the initial mapping phase, and not during tracking.

The underlying algorithm for building the map is described in detail by Algorithm 1. To summarize, when the the proximity sensor indicates that the user has encountered a new doorway, we add two vertices, one directed edge, and one undirected edge to the graph. The directed edge is augmented with the current proximity sensor reading for the doorway, while the undirected edge is augmented with the counted number of footsteps since the previous doorway. The only other case involves passing through a previously mapped doorway in a different direction, in which case the algorithm proceeds in the same manner, except that the two vertices representing the doorway already exist and do not need to be added to the graph. In this way, the graph can be constructed incrementally in real time.

Note that Algorithm 1 deals only with the case where each doorway is visited twice with exactly one trip in each direction, i.e. tree-structured environments with allowances for simple loops. Relaxing this assumption leads to the possibility of doorway ambiguities that require additional doorway-identifying input from the user (or a non-deterministic map-building approach). Note also that learning a true distribution

$\{\mu_E, \sigma_E\}$over proximity sensor readings for each doorway requires walking through the map more than once to have multiple instances of training data for each doorway.

4.2 Localization

Once we have constructed this map, we use a particle filter [11] to track the user's movements across the edges of the graph. Bayesian filtering is a traditional approach to the localization problem in which, at time t, we recursively estimate the posterior distribution $P(X_t|Z_{1:t}, U_{1:t-1})$ of some state X_t (the location of the user) conditioned on all measurements $Z_{1:t}$ (door detections from proximity sensor readings) and control data $U_{1:t-1}$ (footstep estimates from accelerometer data) up to times t and $t-1$ respectively as:

$$P(X_t|Z_{1:t}, U_{1:t-1}) = kP(Z_t|X_t) \int_{X_{t-1}} P(X_t|X_{t-1}, U_{t-1})P(X_{t-1}|Z_{1:t-1}, U_{1:t-2})$$

We call $P(Z_t|X_t)$ the *measurement model* and $P(X_t|X_{t-1}, U_{t-1})$ the *motion model*.

In a particle filter formulation we maintain a number of samples, each of which is a hypothesis about where the user is in the environment. Taken together, this set of samples approximates the probability distribution $P(X_t|Z_{1:t}, U_{1:t-1})$ over the user's current state. Each sample $x^{(i)} = \{e^{(i)}, r^{(i)}, d^{(i)}\}$ indicates a position on the map in terms of the current edge $e \epsilon E$, a distance r along that edge, and a direction of motion $d \epsilon \{-1, 1\}$. To implement tracking with the particle filter, we iteratively update each sample's state with the motion model, weight each sample according to the measurement model, and resample a new set of particles according to these weights.

Motion Model. The motion model $P(X_t|X_{t-1}, U_{t-1})$ is used to predict where the user will be in the next time step, given the previous state X_{t-1} and our accelerometer readings U_{t-1}. In sampling from the motion model we move each particle $x_{t-1}^{(i)}$ by adjusting the value of $r_{t-1}^{(i)}$ according to a Normal distribution $\mathcal{N}(\mu = d_{t-1}^{(i)} * u_{t-1}, \sigma = 0.1)$, where u_{t-1} is the number of footsteps sensed since the last time step and $d_{t-1}^{(i)}$ is the current direction of the particle. If $r_t^{(i)} > r_{e_{t-1}^{(i)}}$ or $r_t^{(i)} < 0$ then the particle has moved off of the previous edge $e_{t-1}^{(i)}$ and a new edge $e_t^{(i)}$ is assigned by sampling uniformly from all edges adjacent to the vertex which the particle overstepped. Finally, if the sample is on an undirected edge, we reverse its direction with some small probability according to $P(d_t^{(i)} = -d_{t-1}^{(i)}) = 0.1$.

Measurement Model. The map we have constructed makes the measurement model $P(Z_t|X_t)$ particularly simple. At each time step we get a measurement Z_t which is the pulse length of the proximity sensor's currently observed doorway, or zero if there is no doorway present. We simply evaluate the current doorway reading with respect to the probability distribution that lives on each sample's current edge, i.e. $P(Z_t|x_t^{(i)}) = \mathcal{N}(\mu_{e_t^{(i)}}, \sigma_{e_t^{(i)}})$ and weight each sample by this amount. Intuitively, if the motion model propagates a particle onto a directed doorway edge, it will only survive the resampling process if the sensor's current reading truly indicates passage through that doorway at the current time step. Thus, particles are kept from escaping through doorways too early, but are free to roam from edge to edge within the same room.

5 Results

To determine the feasibility of solving the tracking task with our limited sensor data, we first evaluated the performance of each sensor individually. Note that rather than doing computation on the wearable device, in these experiments the device communicated wirelessly through BlueTooth to a stationary PC within range.

We tested the accuracy of the footstep detector in a home environment. Over several minutes of data totaling 418 footsteps, the system correctly identified 407 footsteps for an accuracy of 97.4%. In these tests, no effort was made to alter walking patterns in any way to improve accuracy.

In an office setting, we designed an obtacle course involving a mixture of walking through doorways and performing finger gestures. Each run of the course involved 5 doorways and 7 distinct finger gestures (the gestures themselves were the numbers 1 through 7). The system recognized 58 out of 60 events correctly, including identifying the correct number of fingers used in gesturing, for a total of 96.7% accuracy, with one false positive due to a human walking too closely past the user, a clear weakness of the current system. Out of the 5 runs, the final 4 had 100% accuracy, suggesting that a user can learn to correct his or her errors over time while using the device.

Given that the sensors were sufficiently accurate for our tasks, we tested both mapping and localization in a home environment. An apartment consisting of five rooms with four doorways was chosen for testing. As expected, the results improved as a user became more experienced with the sytem. Thus, to illustrate the results of our tests, we present a representative mapping and tracking result for a single experienced user. For the mapping task, the user was able to traverse all the rooms in under 35 seconds at a normal walking pace and by pausing only to label each doorway with a finger gesture before moving through the next doorway. The data from the device was sent wirelessly via BlueTooth in realtime to a desktop computer in the same apartment. Through this process, a topologically-correct augmented map with accurate footstep measurements was successfully constructed (see Figure 5).

Finally, using this map just after it was constructed, the user was successfully tracked walking around all five rooms, going through doorways on eight separate occasions. 500 particles were used during tracking. Figure 6 shows the distribution of particles at six time slices in the tracking sequence. When no doors are observed, the distribution grows uncertain, and the particles spread out. When a door is sensed, the distribution becomes sharply peaked in one or two locations, and the particles are more tightly bunched. The weighted mean location of particles amounts to a hypothesis about which room on the map is being occupied. Accordingly, the tracking accuracy was 100% for the results shown here from an experienced user of the system.

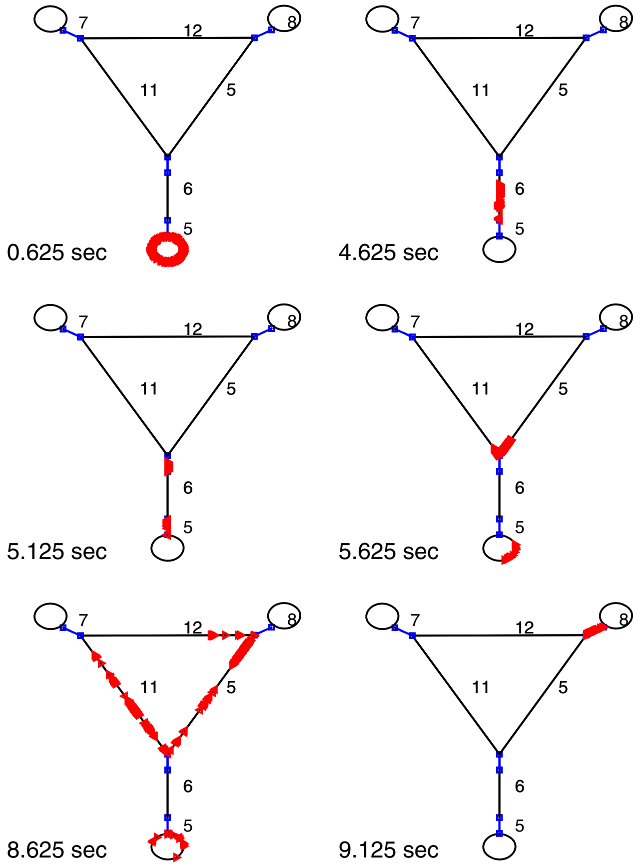


Fig. 6. Tracking results. As the subject passes through three doorways in succession, the motion model propagates particles along edges according to detected footsteps, and each particle is weighted by comparing observed and expected sensor readings.

6 Future Work and Conclusions

Weaknesses of our method include the reliance on significant differences between doorway sensor readings and the effective inability to recover from a tracking failure at any time when a doorway is not being sensed. These problems can be addressed by replacing the binary proximity sensor with one or more analog proximity sensors to obtain more detailed doorway measurements. In addition, since the sensors are worn on the body, the performance of the system depends upon the behavior and skill of the user. For example, at present, the size of a doorway is dependent upon the speed with which it is passed through. A user who is unable to maintain a constant speed across multiple runs will not be tracked as well as a more consistent walker. It is our hope that using additional accelerometer data can alleviate this problem.

The significance of our contribution here is two-fold. First, we have shown that an inexpensive wearable device using small amounts of sensor data can be used for quick and easy topological *mapping* of completely unknown environments. Second, we have demonstrated success in *tracking* users across these automatically constructed maps. Most importantly, we have illustrated the feasibility of cheap, lightweight, wearable mapping and localization devices.

References

1. I. Choi and C. Ricci. Foot-mounted gesture detection and its application in virtual environments. In *IEEE International Conference on Systems, Man and Cybernetics*, 1997.
2. H. Choset. *Sensor based motion planning: The hierarchical generalized voronoi graph*. PhD thesis, California Institute of Technology, 1996.
3. H. Choset and K Nagatani. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *IEEE Trans. on Robotics and Automation*, 17(2), April 2001.
4. S-W. Lee and K. Mase. Incremental motion-based location recognition. In *ISWC*, 2001.
5. S-W. Lee and K. Mase. Activity and location recognition using wearable sensors. *Pervasive Computing*, 1(3), July-Sept 2002.
6. L.Liao, D.Fox, J.Hightower, H.Kautz, and D.Schulz. Voronoi tracking: Location estimation using sparse and noisy sensor data. In *IROS*, 2003.
7. C. Metzger, M. Anderson, and T. Starner. Freedigiter: A contact-free device for gesture control. In *ISWC*, 2004.
8. C. Randell and H. Muller. Context awareness by analysing accelerometer data. In *ISWC*, 2000.
9. A. Ranganathan and F. Dellaert. Inference in the space of topological maps: An mcmc-based approach. In *IROS*, 2004.
10. R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *IJCAI*, 1995.
11. S. Thrun, D. Fox, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, January 2001.