The Mobile Sensing Platform: An Embedded Activity Recognition System

The MSP is a small wearable device designed for embedded activity recognition with the aim of broadly supporting context-aware ubiquitous computing applications.

Tanzeem Choudhury
Dartmouth College
Sunny Consolvo, Beverly Harrison, Jeffrey Hightower, Anthony LaMarca, Louis LeGrand, Ali Rahimi, and Adam Rea
Intel Research
Gaetano Borriello, Bruce Hemingway, Predrag "Pedja" Klasnja, Karl Koscher, James A. Landay, Jonathan Lester, and Danny Wyatt
University of Washington
Dirk Haehnel
Stanford University

Activity-aware systems have inspired novel user interfaces and new applications in smart environments, surveillance, emergency response, and military missions. Systems that recognize human activities from body-worn sensors can further open the door to a world of healthcare applications, such as fitness monitoring, eldercare support, long-term preventive and chronic care, and cognitive assistance. Wearable systems have the advantage of being with the user continuously. So, for example, a fitness application could use real-time activity information to encourage users to perform opportunistic activities. Furthermore, the general public is more likely to accept such activity recognition systems because they are usually easy to turn off or remove.

For systems implementing these applications to be practical, the underlying recognition module must detect a variety of activities that are performed routinely in many different manners by different individuals under different environmental conditions. This presents the challenge of building systems that can handle the real world's noisy data and complexities. Furthermore, deploying the systems imposes some important constraints. The deployment must protect the user's privacy as well as the privacy of those with whom the user comes in contact. The sensors must be lightweight and unobtrusive, and the machine-learning algorithms must be trainable without requiring extensive human supervision. These constraints have made robust recognition systems difficult to engineer.

Over the past four years, we've been building an automatic activity recognition system using on-body sensors. The Mobile Sensing Platform (MSP) tackles several of these design and deployment challenges. Moreover, we've carried out several real-world deployments and user studies, using the results to improve the hardware, software design, and activity recognition algorithms. The lessons learned have broad relevance to context-aware ubiquitous computing applications.

Activity recognition systems

Activity recognition systems typically have three main components:

- a low-level *sensing module* that continuously gathers relevant information about activities using microphones, accelerometers, light sensors, and so on;

Published by the IEEE CS ■ 1536-1268/08/$25.00 © 2008 IEEE

- a *feature processing and selection module* that processes the raw sensor data into features that help discriminate between activities; and
- a *classification module* that uses the features to infer what activity an individual or group of individuals is engaged in—for example, walking, cooking, or having a conversation.

A feature might be low-level information, such as frequency content and correlation coefficients, or higher-level information such as the number of people present. Because human activities are complex and sensor signals have varying amounts of noise, classification algorithms are almost always probabilistic.

The MSP system architecture also consists of these three activity recognition components. However, the MSP evolved in an iterative process that revealed a core set of component requirements after several real-world deployments. The presentation here of our development process includes lessons learned at each stage and how the lessons contributed to our current system.

Hardware platform v1.0: Wireless multimodal sensing

Many recent wearable systems for activity recognition place a single type of sensor, typically accelerometers, in multiple locations (anywhere from two to 12) on the body.[1,2] However, this approach's obtrusive usage model has limited its mass adoption. In addition, its use of a single sensor type restricts the range of activities it can recognize—for example, accelerometers are mainly useful for inferring a limited set of physical activities.

An alternate approach is to use multiple sensor types—that is, multimodal sensors—and collect data from a single body location. Some older research in activity and context recognition explores this approach.[3] Recent studies have shown that the information gained from multimodal sensors can offset the information lost when sensor readings are collected from a single location.[4,5] The sensors' complementary cues are also useful for recognizing a wider range of activities. For example, an accelerometer and audio together can detect whether the user is sitting versus sitting and watching TV.

For wide-scale adoption of activity recognition systems, we hypothesized the need for a sensing platform that

- packages multimodal sensors into a single small device,
- avoids using physiological sensors or sensors that require direct contact with the skin, and
- either integrates into a mobile device, such as a cell phone, or wirelessly transmits data to an external device.

Originally, we assumed the external device would log and process the sensor streams, so the sensing platform could have limited processor capability and no local storage.

To better understand the usefulness of different sensor modalities in inferring human activities, we designed and built a multimodal sensor board that simultaneously captured data from seven different sensors (see figure 1). We selected the sensors for their general usefulness (as evidenced by related work in activity inference),[4,6,7] small footprint, and low power consumption. The sensor board attached to Intel's iMote, a 32-bit ARM7-based wireless node, which communicated with handheld devices, desktop computers, and cell phones via its Bluetooth RF Communication (RFCOMM) protocol, USB cable, or a compact flash bridge.

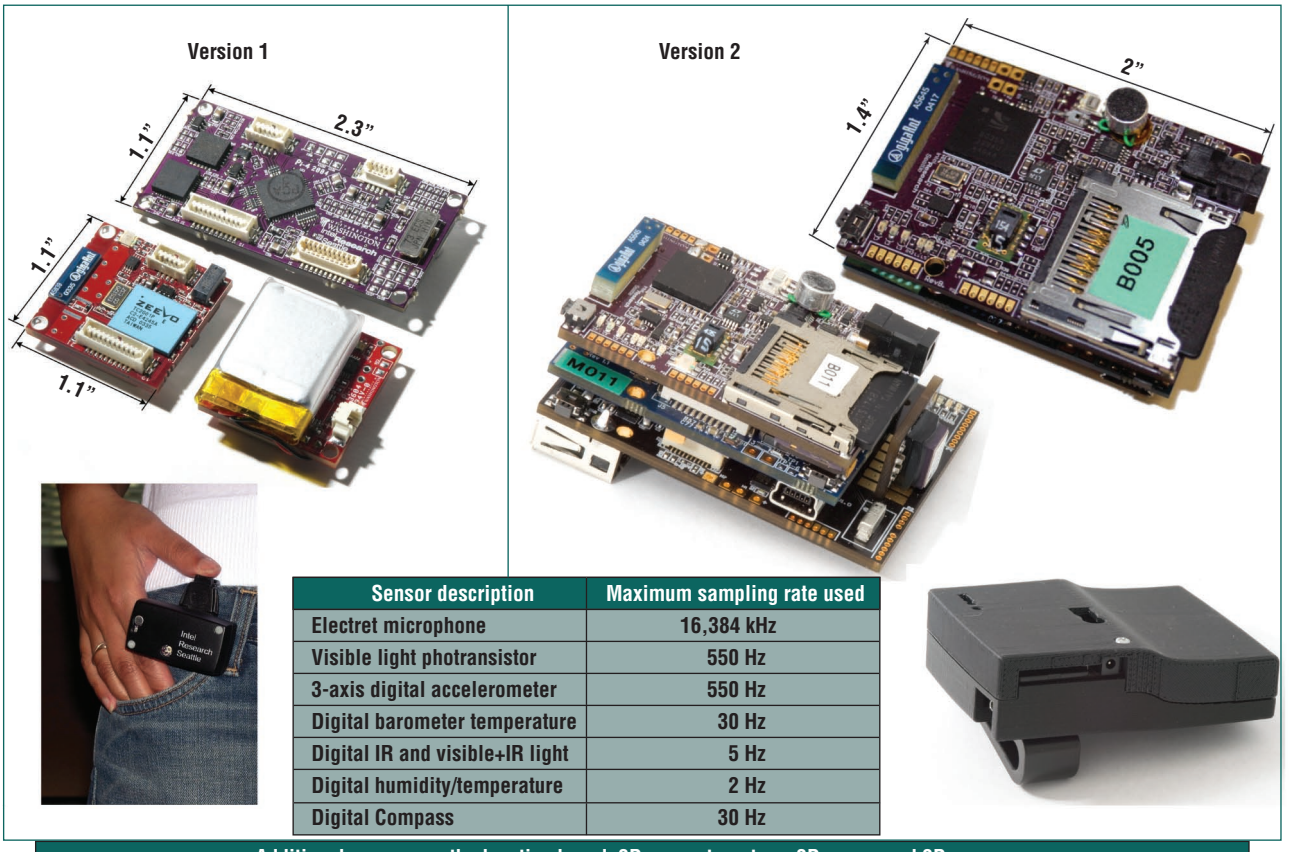
The device was small and lightweight enough (1.52 oz/46 g) to wear comfortably for long periods of time. With all the sensors running continuously, the platform's first version consumed approximately 43 mW of power. It could run for more than 12 hours on a 200 mAh Li-Polymer battery. However, when streaming data over Bluetooth to a cell phone, the battery only ran for about four hours.

Deployments

Our initial deployments focused on gathering real-world activity traces for training activity classifiers. We recruited 15 volunteers to wear the MSPs as they went about day-to-day activities, such as walking, climbing stairs, cooking, working on a computer, and so on. The volunteers generated over 50 hours of data, which we collected over eight noncontiguous weeks.[5,8,9] We based our activity selection on two factors: application scenarios that interested us—specifically, those that encourage physical activity and support eldercare—and prior work in activity recognition systems. Focusing on activities already studied in existing systems helped us compare our system's performance with that of others.

In addition, we conducted a larger, longitudinal deployment to gather data on group interactions and face-to-face social networks.[10] For this deployment,

> The MSP system architecture evolved in an iterative process that revealed a core set of activity recognition component requirements.



Sensor description	Maximum sampling rate used
Electret microphone	16,384 kHz
Visible light photransistor	550 Hz
3-axis digital accelerometer	550 Hz
Digital barometer temperature	30 Hz
Digital IR and visible+IR light	5 Hz
Digital humidity/temperature	2 Hz
Digital Compass	30 Hz

Additional sensors on the location board: 3D magnetometers, 3D gyros, and 3D compass

Figure 1. The Mobile Sensing Platform hardware, versions 1.0 and 2.0., with and without cases. Both MSP versions support seven sensors. In addition, version 2.0 offers a location daughterboard that includes additional 3D sensors. The MSP can communicate wirelessly with other devices in real time or store raw sensor data, features, or activity recognition results for offline use.

we recruited 24 graduate student volunteers and gathered data on them simultaneously for one week per month over the course of a year. The result was more than 4,400 hours of sensor data that we're currently analyzing offline.

Lessons learned

Packaging the multimodal sensors into a small form factor was an appealing characteristic of the version 1.0 MSP platform. Unfortunately, depending on an external device for data processing and logging proved to be a problem even before any serious data collection began.

Communication, storage, and processor issues. The Bluetooth connectivity wasn't reliable enough to continuously stream sensor data (including audio at 8 or 16 kHz). The packet losses and intermittent connection drops required us to switch to a wired solution where the sensor board was physically attached to a PDA via a USB cable.

We used this "sensor plus iPAQ" combo to collect the data sets we've described. This solution worked as a temporary research prototype, but it clearly wasn't feasible longer term because it required participants to carry a bulky, wired device combination simply to collect data. We might have been able to mitigate the drops by implementing a standard transport protocol instead of relying on Bluetooth's RFCOMM stack. However, this would provide only a partial fix to the problems we encountered during data collection.

Our initial deployments showed that storing and processing data locally would significantly increase the data quality (no packet loss) and recording duration (via compression), while reducing the physical burden on the participant. Additionally, supporting interactive applications that react according to a given activity or context called for computational power sufficient to classify sensor traces into usable activity or context labels in real time. Some behavior and health-monitoring applications might also need to store the raw sensor data or inferred behavior statistics locally for additional offline processing to address longer-term trends or to infer more complex human behavior using models.

Privacy. The larger group deployment reenforced the importance of considering privacy aspects of data logging. Collecting sensor data, particularly from a microphone, involves recording people in unconstrained and unpredictable situations, both public and private. The results can include recorded information about uninvolved parties without their consent—a scenario that, if raw audio is involved, is always unethical and often illegal.

We therefore needed the iPAQ's computational resources to process the raw audio data on the fly and record useful features. For example, the system needed to record enough information to infer that a conversation had occurred but not enough to reconstruct the words that were spoken.[10]

Battery life. Several other problems arose from our unconventional use of the PDA as a mobile sensor node. The PDA is optimized for sporadic, short tasks, which is not how we were using it. As a result, the PDA batteries didn't last as long as we had anticipated, often keeping us from reaching our goal of collecting 8 to 10 hours of data per day.

Participants found it easy to recharge their PDAs overnight but would often forget to change the battery midday. So, battery life sufficient to support data collection and processing during waking hours (about 12 to 16 hours) was a fundamental usability issue.

Sensing and inference module. Not surprisingly, mobile devices are often optimized to support a specific usage model. Using these devices for logging or inference can significantly reduce the battery life and computational resources, thereby compromising the device's usability. Furthermore, integrated solutions that package sensors into a mobile device might not provide all the flexibility an activity recognition system needs.

Depending on the activities being modeled and the users' preferences, the system and its sensors might need to be located away from where the user carries his or her mobile device. As a result, it might be preferable for the system to operate as a standalone device with its own CPU and battery for sensing and inference. The device could then communicate with other mobile devices as needed without taxing their resources or limiting their usability.

Location sensing. The different MSP sensors were sufficient for recognizing the range of daily activities and interactions that interested us. However, geographical location information has proved helpful in activity recognition.[11] For example, it's unlikely that a user in the middle of a lake is riding in an elevator. We therefore sometimes used an external GPS unit or scanned for Wi-Fi access points using an iPAQ or cell phone to obtain location. In some cases, knowing just the location is enough to infer coarse-level activities. Integrating location sensing into the platform would enable us to support a wider range of context-aware applications.

> The system needed to record enough information to infer a conversation, but not enough to reconstruct the words spoken.

Hardware platform v2.0: Local storage, better processor and battery life

We developed the second-generation MSP to address the major shortcomings identified from the version 1.0 deployments to prepare for its use in several in situ user studies.

As in version 1.0, version 2.0 has a sensor board equipped with six of the previously outlined seven sensors (see figure 1). We moved the compass to a separate daughterboard, which we designed to support experiments in location and inertial sensing. Version 2.0's optional daughterboard provides 3D magnetometers, 3D gyros, a 3D compass, and a USB host. This option increased the MSP's power usage (20 percent) as well as its size.

To support future MSP extensions with new sensors, version 2.0 included additional serial connections and general-purpose I/O pins. The sensor board includes a removable miniSD card, which bounds the storage size—we currently use 2-Gbyte cards. The board also includes a Bluetooth radio that can communicate in both the RFCOMM and personal-area-network (PAN) profiles. This capability lets MSP both connect to IP networks via Bluetooth access points and pair with devices such as cell phones and PDAs.

We attached the sensor board to the iMote2 (the second generation of Intel's iMote). The iMote2's onboard 416 MHz Xscale processor enables feature-processing and activity-recognition algorithms to run in real time on the device. The second-generation MSP runs on an 1800-mAh-hour battery and is small enough to wear on a belt clip. It weighs only 4.06 oz (115 g) and measures 2 inches on its longest dimension. Depending on the computing and communication workload, the version 2.0 MSP runs between 10 and 20 hours on a single charge.

Data processing and inference

Activity recognition requires sensor data

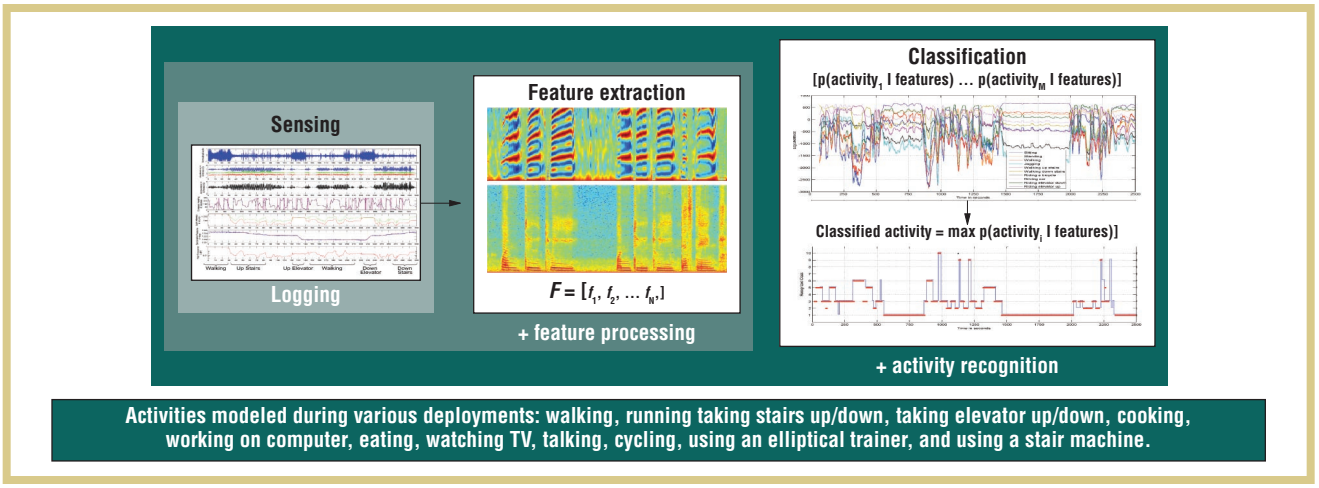


Figure 2. MSP software flow diagram for the classification system. The MSP has a flexible usage model that includes logging, feature processing, and activity recognition.

to be processed and classified into meaningful activity chunks. This involves extracting various features that serve as input to the classification module. Given a set of observed features, the system must learn the parameters of its algorithms from examples of the relevant activities. This training process is usually done offline. If the system must recognize many activity types, manually training the system would require too much human effort and activity-specific tuning. Thus, both the feature selection and the training process must be automated. We hypothesized that automating feature selection would let us develop an activity recognition system requiring less human effort during training, without sacrificing recognition accuracy.

A feature's usefulness depends on the specific activity to be inferred. For example, frequency information from the three-axis accelerometer is important in determining activities such as walking, running, and related gaits. Periodicity from the microphone's auditory signal is useful in determining whether someone is talking. Some features might be deterministic transformations of the raw sensor data (for example, frequency content), while others can be probability measures (for example, the user's likelihood of being in a certain location).

The timescale at which features are computed also impacts recognition. For example, human speech is usually analyzed at millisecond resolution, whereas a variety of physical activity models use features computed at 0.1 to 10 Hz. Furthermore, contextual information about behavior is often computed over minutes or even hours (consider cooking or attending meetings).

Certain feature sets will help in recognizing certain activities or classes of activities, but a feature megaset could be useful in detecting a wide range of activities. A classifier or model could then automatically select the feature subset that's most suited for a given task.

Like the MSP hardware, the MSP data processing and inference software has evolved from deployment experience.

Inference v1.0: Automatic feature selection and classification

The pervasive computing community hasn't deeply explored the problem of automatically identifying a small set of features useful for recognizing a given set of activities. Although you can learn activity models from a large set of features without automatic feature selection, it requires a huge amount of training data to estimate the model parameters associated with the features. If you train classifiers using a small quantity of data, the model parameter estimates won't be reliable and the resulting system will be brittle. On the other hand, gathering adequate training data to represent the activities being modeled is often a challenge because it requires considerable human effort. So, the capability to automatically select a small subset of useful features is highly desirable.

We've implemented a feature selection approach based on boosting.[5,12] In our approach, the boosting technique automatically selects a small set of useful features from a possible superset of 600, which includes linear and log-scale frequency coefficients, cepstral coefficients, spectral entropy, band-pass filter coefficients, correlations, integrals, means, and variances.[5] Features are selected iteratively to reduce the classification error rate and train classifiers that can recognize the different activities. The trained system outputs the probability scores for different activities that might be occurring at a given time; then it selects the most probable activity (see figure 2).

In some cases, however, you might know the most useful feature subset in advance. If so, you can use it directly in training the classifier—for example, to detect the presence of a human voice.[10] In these situations, you can omit the feature selection step.

Deployments. We evaluated our sys-

TABLE 1
Inference algorithms' performance for activities classified every 0.25 seconds.*

	Inference v1.0 (temporal information not included, supervised)	Inference v2.0 (temporal information included, supervised)	Inference v3.0† (temporal information included, semisupervised)		
Training data labeled	100%	100%	5%	20%	40%
Accuracy‡	83.6%	93.8%	79.7%	83.1%	87.4%

*Activities: walking, sitting, standing, taking stairs up and stairs down, taking elevator up and down, brushing teeth.
†For the semisupervised approach (v3.0), the columns report performance when different fractions of the training data (5, 20, and 40 percent) are labeled.
‡Detailed performance numbers, including precision and recall, on the complete set of activities are available elsewhere.[5,8,9]

tem's performance on the activity data sets we collected. Table 1 shows the v1.0 sample results in the first data column; detailed performance numbers are available elsewhere.[5] By automatically inferring the features that were most useful, we discovered that two modalities in particular (out of seven) yielded the most discriminative information for our activities—namely, the accelerometer and microphone. These two modalities provide complementary information about the environment and the wearer. The audio captures sounds produced during various activities, whereas the accelerometer data is sensitive to body movement.

Other sensors yielded specific types of information useful for certain activities. For example, the barometric pressure sensor was sensitive enough to detect the activities of riding in an elevator or going up and down stairs.[5] Once we had trained the system using data from a few individuals (usually six to eight), it generalized well to new individuals—that is, recognition rates for new individuals were comparable to those for individuals whose data was part of the training set.

During training, we picked data collectors who represented the age, weight, and gender of the system user population.

Lessons learned. Automatic feature selection helps in recognizing activities from a small feature subset. However, making a decision about activity using features from a single time point fails to take advantage of the temporal continuity or structure present in activities and can result in choppy classification. For example, a single contiguous walking segment might get broken into multiple segments. Choppy classification is a problem for applications that must take action based on the occurrence of a given activity.

Information about the user's context can also improve inference. For example, people seldom drive a car indoors or vacuum outdoors.

Classifiers can take advantage of these temporal/structure and activity/context dependencies, often called *structured prediction*, to improve their performance.

Inference v2.0: Activity structures

To take advantage of activity structures, we developed an extension to the boosting algorithm. The extension performs feature selection and structured prediction in a unified, more efficient way; details of the technique are available elsewhere.[8] In experimental results, the extension improved the MSP's classification smoothness and accuracy by approximately 10 percent (see table 1). In fact, the unified approach outperforms both standard boosting methods and standard temporal or structured-prediction techniques, such as hidden Markov models and conditional random fields.[8]

The inference techniques we've applied so far rely on learning from explicitly labeled examples. This approach, called supervised training, requires each data point to be associated with an activity label—almost always provided by humans. While collecting large amounts of sensor data might not be difficult, accurately labeling large amounts of data is often impractical. Data labeling also has privacy implications, as it can require human observers to review sensitive data, such as video and audio recordings.

Unsupervised training methods that group feature traces in relevant activity clusters automatically, without using labels, have shown some promise in limited settings.[13] However, fully unsupervised techniques don't know what activities to model, so they run the risk of grouping data into clusters that don't correspond to the activity patterns the application needs.

Inference v3.0: Reductions in labeled training data

If we can train activity recognition systems using a few labeled examples and all the available unlabeled data (often referred to as semisupervised training), system performance is likely to improve compared to using supervised methods on a limited amount of labeled data. Semisupervised techniques are also useful for personalizing activity models to a specific user's behavior, as the system can adapt the model parameters according to the user's unlabeled data, even after system deployment.

To reduce the dependency on labeled data, we developed a semisupervised version of our previous algorithm. The new algorithm can select features, use activity structures, and take advantage of available unlabeled data[9] (see table 1 for experimental results). The new algorithm is also computationally very efficient, requiring significantly less training time and memory compared

TABLE 2
Current Mobile Sensing Platform implementations.

Component	MSP hardware v1.0	MSP hardware v2.0
Processor	ATMega128 microcontroller on sensor board ARM7 processor on iMote	ATMega128 microcontroller on sensor board PXA271 Xscale processor on iMote
Storage	No onboard storage	miniSD card slot (current storage 2 Gbytes)*
Communication	Bluetooth radio RFCOMM	Bluetooth radio (both RFCOMM and PAN), plus Zigbee radio
Battery life	200 mAH Li-Polymer battery: • Basic data handling: 37 mA • Stream data: 50 mA	1800 mAH Li-Polymer battery: • Basic data handling: 103 mA • Log data: 127 mA • Stream features: 118 mA • Stream inference results: 113 mA
Inference	No onboard inference capability	Embedded inference software version 1.0

*Records over 10 days of audio features plus other sensor traces sampled at the highest rate used in any of our experiments.

to related semisupervised methods. In our experiments, the training time decreased by a factor of 20 and the memory requirement by a factor of 13 (details are available elsewhere[9]). We believe these efficiencies will appeal to application developers.

Implementation status

Table 2 summarizes the current MSP implementations. The embedded software runs inference version 1.0 with partial support for temporal models. We plan to port inference versions 2.0 and 3.0 into the platform in the near future. Although these inference models will require more computational resources, the requirements are within the MSP's capabilities.

The computational cost of training the models developed as part of inference versions 2.0 and 3.0 is significantly higher. However, training is often done offline, and the embedded software runs the inference using the trained model parameters. The MSP software processes the sensor streams, computes the features, and applies the trained models to classify activities.

The MSP runs standard Linux with complete support for multitasking, IP networking via Bluetooth or USB, and a flash file system. We've developed a flexible feature extraction and classification runtime library that uses a simple XML configuration file for specifying a sensor set, sampling rates, and algorithms. The MSP can stream the resulting computations to storage over the network or to a custom program running on the MSP.

To improve MSP program development, we've configured a Linux VMware image with the MSP cross-development environment.

Real-world deployments and applications

We've used MSP version 2.0 hardware in a variety of pilot deployments and experiments. These deployments have reinforced some of the lessons learned during the design iterations and validated the MSP's overall flexibility in supporting different usage models. A brief sampling of the research deployments show different ways the platform could be used.

UbiFit Garden field study

UbiFit Garden is an application that uses on-body sensing, real-time activity inference, and a mobile ambient display to encourage individuals to be physically active. UbiFit Garden uses the MSP to infer five types of activities: walking, running, cycling, using an elliptical trainer, and using a stair machine. The MSP communicates the inferred activity labels via Bluetooth to a mobile phone that applies heuristics to smooth the inferences into contiguous activity chunks, such as a 47-minute walk. When the individual starts an activity, a flower blooms in a garden on the background screen of his or her mobile phone.[14] Once we've incorporated inference version 2.0 into the embedded software, this smoothing can happen automatically on the MSP, thus reducing Bluetooth communications and increasing battery life.

Our first in situ study of the UbiFit Garden system was a three-week field trial (N = 12) conducted during the summer of 2007. All participants were volunteers who regularly used mobile phones and wanted to increase their physical activity level. Participants understood that the study involved wearing a fitness device. They represented a variety of mostly nontechnical occupations including homemakers, a marketing specialist, receptionist, elementary school employee, musician, copywriter, filmmaker/videographer, professional actor/dancer, director of external affairs for a nonprofit agency, and software implementer.

Participants received a user manual explaining how to use the MSP, phone, and UbiFit Garden application. The research team spent about 30 minutes at the beginning of the study going over the manual; they interacted with

participants only during scheduled interview sessions (at the beginning, after the first week, and at the end of the three weeks) or if participants reported a technical problem with the device (two reports total).

During the study, the MSP inferred a total of 207 instances of physical activity events for the 12 participants. Participants left most (77 percent) of the MSP-inferred events unedited in UbiFit's fitness log, which ran on the mobile phone. Response to UbiFit Garden was overwhelmingly positive. At the study's end, most participants said they would prefer a system that included activity inference, such as the MSP, over one that relied solely on manual entry (study details are available elsewhere[14]).

Recall accuracy study

The ability to accurately recall daily activities is central to many disciplines. In health sciences, establishing health benefits from physical activity is primarily done on the basis of self-report data, typically surveys asking people to recall what physical activity they performed in the last week or two.[15] However, data suggests that, especially for frequent activities, people tend to overreport physical activity and underreport sedentary behaviors. We wished to systematically investigate how we might minimize participant burden of self reporting, while maximizing the accuracy of the information reported.

In early 2007, we conducted a study to understand how well people recall the amount of their active behaviors, such as walking, and their sedentary behaviors, such as sitting. Twenty participants volunteered to wear an MSP and carry an associated cell phone for eight typical workdays (approximately 12 hours per day). Participants represented a variety of professional profiles including office workers, retail workers, homemakers, students, a dancer, a tight rope instructor, and a waiter. The office workers were researchers, engineers, a public health professional, and an environmental scientist.

The cell phone randomly surveyed participants throughout the day at varying frequencies—ranging from once a day to every 20 minutes—to ask about their walking and sitting patterns. To score participants' responses, we used the MSP to estimate how much walking and sitting they did during the period in question. We used the MSP's activity recognition data as "ground truth," so we trained the classifiers specifically to maximize the recognition accuracy for walking and sitting, which we benchmarked independently at 96 percent and 93 percent, respectively.

Other ongoing studies

Several other studies using the MSP are currently under way. One project involves developing algorithms to compute accurate caloric expenditure from user activities, which we will use to build an ambient personal-fuel gauge.

Another project is targeting Type I diabetics to adjust insulin dosages on the basis of real-time activity information and thus prevent dangerous hypoglycemic episodes.

A third project couples MSP with GPS information to better understand how urban planning affects public health by measuring outdoor physical activity levels in different types of built environments.

Our studies identified three critical capabilities for mobile inference systems to support a variety of usage scenarios and applications:

- *Form factor*. The sensing device must be small and unobtrusive yet capable of recognizing a broad range of activities. (Our current prototype is still too large for commercial use, but study participants understood that the device would eventually be smaller.) A single multimodal device, as opposed to multiple separate sensor nodes, is likely to gain greater user acceptance.
- *Sensing hardware*. The system needs enough storage for data logging, sufficient computational resources for data processing and running simple inference algorithms, wireless connectivity to drive interactive applications, and enough battery power to run for at least an entire day. Systems that store data on a device used for other purposes, such as a mobile phone, must accommodate those other storage needs as well (for example, leaving room for saving photos).

Most UbiFit Garden participants said they would prefer an activity-inference system over one that relied solely on manual entry.

- *Recognition software*. In addition to being computationally efficient, the recognition algorithms should minimize the amount of human effort required during training and data labeling. It should be easy to extend the algorithms to recognize a wide range of activities and to personalize to a given user's data.

In developing the MSP, we ad-

dressed key challenges in building a scalable, mobile, activity-inference system. We will continue to address some of these remaining challenges in our ongoing research. For example, these systems must be able to recognize multiple parallel activities simultaneously and deal with interleaving activities, such as a situation where a user starts to cook, is interrupted by a phone call, and then returns to cooking. They must automatically learn new activities, intelligently prompt the user for new activity labels (by maximizing the information gained from a minimal number of prompts), and use these labels to build models for new activity types or to improve the existing models. P

ACKNOWLEDGMENTS

We thank David Wetherall for his valuable feedback on the article. The US National Science Foundation partially supported this work under grant IIS 0433637.

REFERENCES

1. L. Bao and S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Proc. 2nd Int'l Conf. Pervasive Computing* (Pervasive 04), LNCS 3001, Springer, 2004, pp. 1–17.

2. N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor Activity Context Detection for Wearable Computing," *Proc. 1st European Symp. Ambient Intelligence* (EUSAI 03), LNCS 2875, Springer, 2003, pp. 220–232.

3. A. Schmidt et al., "Advanced Interaction in Context," *Proc. 1st Int'l Symp. Handheld and Ubiquitous Computing* (HUC 99), LNCS 1707, Springer, 1999, pp. 89–101.

4. U. Maurer et al., "Location and Activity Recognition Using eWatch: A Wearable Sensor Platform," *Ambient Intelligence in Every Day Life*, LNCS 3864, Springer, 2006, pp. 86–100.

5. J. Lester, T. Choudhury, and G. Borriello, "A Practical Approach to Recognizing Physical Activity," *Proc. 4th Int'l Conf. Pervasive Computing* (Pervasive 06), LNCS 3968, Springer, 2006, pp. 1–16.

6. P. Lukowicz et al., "WearNET: A Distributed Multi-sensor System for Context Aware Wearables," *Proc. 4th Int'l Conf. Ubiquitous Computing* (Ubicomp 02), LNCS 2498, Springer, 2002, pp. 361–370.

7. S. Park et al., "Design of a Wearable Sensor Badge for Smart Kindergarten," *Proc. 6th Int'l Symp. Wearable Computers* (ISWC 02), IEEE CS Press, 2002, pp. 231–238.

8. L. Liao et al., "Training Conditional Random Fields Using Virtual Evidence Boosting," *Proc. Int'l Joint Conf. Artificial Intelligence* (IJCAI 07), 2007; www.ijcai.org/papers07/Papers/IJCAI07-407.pdf.

9. M. Mahdaviani and T. Choudhury, "Fast and Scalable Training of Semi-supervised Conditional Random Fields with Application to Activity Recognition," *Proc. Neural Information Processing Systems* (NIPS 07), 2007; http://books.nips.cc/papers/files/nips20/NIPS2007_0863.pdf.

10. D.T. Wyatt, T. Choudhury, and H. Kautz, "Capturing Spontaneous Conversation and Social Dynamics: A Privacy-Sensitive Data Collection Effort," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing* (ICASSP 07), IEEE Press, 2007, pp. IV-213–IV-216.

11. L. Liao, D. Fox, and H. Kautz, "Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields," *Int'l J. Robotics Research*, vol. 26, no. 1, 2007, pp. 119–134.

12. R.E. Schapire, "A Brief Introduction to Boosting," *Proc. 16th Int'l Joint Conf. Artificial Intelligence* (IJCAI 99), Morgan Kaufmann, 1999, pp. 1401–1406.

13. A. Krause et al., "Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing," *Proc. 7th IEEE Int'l Symp. Wearable Computers* (ISWC 03), IEEE CS Press, 2003, pp. 88–97.

14. S. Consolvo et al., "Activity Sensing in the Wild: A Field Trial of UbiFit Garden," to be published in *Proc. ACM SIGCHI Conf. Human Factors and Computing Systems* (CHI 08), ACM Press, 2008.

15. J.F. Sallis and B.E. Saelens, "Assessment of Physical Activity by Self-Report: Status, Limitations, and Future Directions," *Research Quarterly for Exercise and Sport*, vol. 71, no. 2, 2000, p. 1–14.

the AUTHORS

Tanzeem Choudhury is an assistant professor of computer science at Dartmouth College. She joined Dartmouth in 2008 after four years at Intel Research Seattle, where she initiated the MSP-based activity-recognition project jointly with Gaetano Borriello. Her research involves developing machine-learning techniques for systems that can reason about human activities, interactions, and social networks in everyday environments. She received her PhD from the Media Laboratory at the Massachusetts Institute of Technology. Contact her at tanzeem.choudhury@dartmouth.edu.

Gaetano Borriello is a professor of computer science and engineering at the University of Washington. He also founded Intel Research Seattle, where he launched the lab on applications of ubiquitous computing technology to healthcare and eldercare, in particular. His research interests include location-based systems, sensor-based inferencing, and tagging objects. He received his PhD in computer science from the University of California, Berkeley. He's an associate editor in chief of *IEEE Pervasive Computing*. Contact him at gaetano@cs.washington.edu.

Sunny Consolvo is a member of the research staff at Intel Research Seattle and a PhD candidate at the University of Washington's Information School. Her research interests include applying user-centered design to ubiquitous computing. In particular, she's interested in the social implications of ubiquitous computing technologies. Her current focus is on developing persuasive technologies to encourage people to incorporate regular and varied physical activity in their everyday lives. Contact her at sunny.consolvo@intel.com.

Dirk Haehnel is an associate research engineer in the Department of Computer Science at Stanford University. His research interests lie in mobile robot navigation and probabilistic methods for state estimation. He received his PhD in computer science from the University of Freiburg, Germany, and worked at the Intel Research Lab Seattle. Contact him at haehnel@stanford.edu.

Beverly Harrison is a senior scientist at Intel Research Seattle. Her research interests include the design and evaluation of novel mobile and sensor-based technologies for ubiquitous computing applications and innovative interaction techniques in pervasive computing applications. She received her PhD in human factors engineering from the University of Toronto. Contact her at beverly.harrison@intel.com.

Bruce Hemingway is a lecturer in the University of Washington's Department of Computer Science & Engineering and manager of its Baxter Computer Engineering Laboratory. His research interests include the development of autonomous computer music generators that will sense mood from physiological indicators and produce music pleasing to the listener. He received an AB in music from Indiana University. Contact him at bruceh@cs.washington.edu.

Jeffrey Hightower is a member of the research staff at the Seattle Lab of Intel Research. His research interests are in devices, services, sensors, and interfaces that help computing calmly fade into the background of daily life. Specifically, he investigates the technology and use of sensor-enhanced mobile computing devices. He received his PhD in computer science and engineering from the University of Washington. He's a member of the ACM and IEEE. Contact him at jeffrey.r.hightower@intel.com.

Predrag "Pedja" Klasnja is a third-year student in information science at the University of Washington. His research interests include ubiquitous computing and HCI, particularly the application of mobile devices in healthcare. He has a BS in physics from the University of Tennessee, Knoxville. Contact him at klasnja@u.washington.edu.

Karl Koscher is a first-year graduate student in computer science at the University of Washington. His research interests include systems security, ubiquitous computing, and intersections of these areas. He received his BS in computer science from the University of Washington. Contact him at supersat@cs.washington.edu.

Anthony LaMarca is the associate director of Intel Research Seattle. His research interests include location technologies, activity inference, ubiquitous computing, and human-centered design. He received his PhD in computer science from the University of Washington. Contact him at anthony.lamarca@intel.com.

James A. Landay is a professor in the Computer Science & Engineering Department at the University of Washington, specializing in HCI. His research interests include automated usability, demonstrational interfaces, ubiquitous computing, design tools, and web design. He received his PhD in computer science from Carnegie Mellon. Contact him at landay@cs.washington.edu; www.cs.washington.edu/homes/landay.

Louis LeGrand is a senior software engineer at Intel Research Seattle. His research interests include inference software for small sensing devices, machine learning, robotics, and unmanned aerial vehicles. He received his MS in electrical engineering from the University of Washington and MS in aeronautical engineering from Iowa State University. Contact him at louis.l.legrand@intel.com.

Jonathan Lester is a PhD student in electrical engineering at the University of Washington. His research interests are computationally aware machine-learning techniques for embedded systems, wearable sensors, and sensor platforms. He received his MS in electrical engineering from the University of Washington. Contact him at jlester@u.washington.edu.

Ali Rahimi is a research scientist at Intel Research Seattle. His research interests are in machine-learning applications to computer vision. He received his PhD in computer science from the Massachusetts Institute of Technology Computer Science and AI Lab. Contact him at ali.rahimi@intel.com.

Adam Rea is a hardware engineer for Intel Research Seattle. He builds small, power-efficient sensing devices. He received his BS in computer engineering from the University of Washington. Contact him at adam.d.rea@intel.com.

Danny Wyatt is a PhD student in the Computer Science and Engineering Department at the University of Washington. His research involves sensing and modeling real-world human social behavior. He received his MS in computer science from the University of Washington. Contact him at danny@cs.washington.edu.

IEEE pervasive COMPUTING
MOBILE AND UBIQUITOUS SYSTEMS

How to Reach Us

Writers
For detailed information on submitting articles, write for our Editorial Guidelines (pervasive@computer.org) or access www.computer.org/pervasive/author.htm.

Letters to the Editor
Send letters to
Shani Murray, Lead Editor
IEEE Pervasive Computing
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
pervasive@computer.org
Please provide an email address or daytime phone number with your letter.

On the Web
Access www.computer.org/pervasive for information about *IEEE Pervasive Computing*.

Subscription Change of Address
Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify *IEEE Pervasive Computing*.

Membership Change of Address
Send change-of-address requests for the membership directory to directory.updates@computer.org.

Missing or Damaged Copies
If you are missing an issue or you received a damaged copy, contact membership@computer.org.

Reprints of Articles
For price information or to order reprints, send email to pervasive@computer.org or fax +1 714 821 4010.

Reprint Permission
To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at copyrights@ieee.org.