VoiceLabel: Using Speech to Label Mobile Sensor Data

Susumu Harada[1], Jonathan Lester[2], Kayur Patel[1], T. Scott Saponas[1],
James Fogarty[1], James A. Landay[1,4], Jacob O. Wobbrock[3]

[1]Computer Science & Engineering
DUB Group
University of Washington

[2]Electrical Engineering
DUB Group
University of Washington

[3]Information School.
DUB Group
University of Washington

[4]Intel Research Seattle
1100 NE 45th Street
Seattle, WA

{harada, kayur, ssaponas, jfogarty, landay}@cs.washington.edu,
{jlester, wobbrock}@u.washington.edu

ABSTRACT
Many mobile machine learning applications require collecting and labeling data, and a traditional GUI on a mobile device may not be an appropriate or viable method for this task. This paper presents an alternative approach to mobile labeling of sensor data called *VoiceLabel*. VoiceLabel consists of two components: (1) a speech-based data collection tool for mobile devices, and (2) a desktop tool for offline segmentation of recorded data and recognition of spoken labels. The desktop tool automatically analyzes the audio stream to find and recognize spoken labels, and then presents a multimodal interface for reviewing and correcting data labels using a combination of the audio stream, the system's analysis of that audio, and the corresponding mobile sensor data. A study with ten participants showed that VoiceLabel is a viable method for labeling mobile sensor data. VoiceLabel also illustrates several key features that inform the design of other data labeling tools.

Categories and Subject Descriptors
H.5.2 [**Information interfaces and presentation**]: User Interfaces – *Voice I/O*

General Terms
Design, Human Factors

Keywords
Data collection, mobile devices, sensors, speech recognition, machine learning

1. INTRODUCTION
Mobile devices continue to mature, decreasing in physical size while gaining new sensing capabilities and reaching new levels of computational power. This provides new opportunities for sensing-based mobile applications to seamlessly integrate into and enhance everyday life. Location-aware devices can fuse information from multiple sources, including GPS, wireless access points, and cell towers to provide reliable location information in both indoor and outdoor settings [4]. Advances in wearable sensing promise to enable a myriad of mobile applications informed by physical activity [12]. Recent work has shown that it is now possible to classify human physical activity using the sensing capabilities of commodity mobile devices such as Apple's iPhone [16]. These advances enable many new mobile applications, including helping people achieve life goals such as maintaining wellness and living in an ecologically sustainable manner [6].

Designing and building sensor-based mobile applications often require inferring high-level classes (e.g., human activities) from low-level sensors (e.g., accelerometers). Although supervised machine learning algorithms have proven a popular and effective approach to such inference [4, 12], training these algorithms requires gathering sensor data, segmenting the sensor data into different classes, and then correctly labeling each class. Learning algorithms then use this accurately labeled data to build a model that can be used to classify a previously unseen and unlabeled set of sensor data (e.g., using accelerometer data to recognize human activity).

Unfortunately, it is often difficult to gather such labeled data necessary for these projects. Collecting labeled training data for human activity recognition on mobile devices can be particularly challenging because the act of providing a label can interfere with the sensing of the activity. Consider the example of interacting with a mobile device's display while performing an activity that is intended to be recognized by sensors attached to or embedded in the device. A person may take his device out of his pocket to press a button for "standing" in order to provide a label for the collected accelerometer data. But by manipulating the device in this manner, the resulting accelerometer "signal" becomes part of the labeled data, thus complicating the use of this data in a machine learning process.

As another approach, a "data collector" can be asked to act freely without concern for manually labeling his activity. A "labeler" can then later segment and label the collected data. For example, a data collector might record a video of his activities in conjunction with the sensor data. The video can then be reviewed by the labeler in order to segment and label the sensor data. Although this avoids problems discussed in the previous paragraph, segmenting and labeling video is time consuming and tiring, making it expensive and error-prone.

We present *VoiceLabel*, a tool that demonstrates a new approach to using speech to gather sensor data labels for training and testing models in activity classification tasks. VoiceLabel spreads the tasks of collecting, segmenting, and labeling data over two components. The first is a mobile tool for collecting sensor data, with which the data collector labels activities by speaking the

ICMI'08, October 20–22, 2008, Chania, Crete, Greece.

name of an activity while performing it. The second is a desktop tool for automatically segmenting recorded data at activity transitions and recognizing the corresponding spoken labels. Labelers can also listen to the recorded labels and verify the accuracy of the speech recognition by interactively browsing a visualization of the collected data and the output of the speech recognizer. Our results show that using voice labeling can be an effective method for collecting labeled data.

The next section briefly reviews related work in the area of mobile data collection and activity recognition. We then further discuss the activity recognition challenge and design considerations that influenced the development of VoiceLabel. We next describe the architecture for our prototype and the final system and the results of two studies. Finally, we discuss the potential to extend the idea of using speech to other labeling tasks and present a brief conclusion.

2. RELATED WORK

Activity recognition systems infer human activities (e.g., walking, jogging, standing) from low level sensors (e.g., accelerometers, barometers, audio) using machine learning techniques [1, 9, 15]. These algorithms learn a model of the relationship between low-level sensor data and classify activities through sets of labeled sensor data. This means that in addition to collecting useful data, a labeler must segment a dataset into chunks and attach some meaningful label to each of these chunks. Errors in these labels make the task of learning useful models more difficult and less robust. Thus, care must be taken to ensure accurate labels.

Researchers have employed several approaches in an effort to gather quality labeled data for learning models to support sensor-based mobile applications. Lester et al. [9] used a sensor platform called the Mobile Sensing Board (MSB) to collect data for activity recognition. In this work, two data collectors wore a small video camera along with the MSB to capture a first-person view of the environment as they performed multiple activities. The data was then annotated by the data collectors who reviewed 12 hours of audio/video footage and labeled each activity. Offline video labeling proved to be a serious bottleneck, and in their later work they collected ground truth data by having an experimenter follow the participant and annotate the data collector's activities using a graphical application on a PDA [12]. While the annotation application reduced some of the workload, it introduced problems. The experimenter had to closely follow the participant, potentially disrupting the course of the activity, while paying close attention to the GUI interface to ensure that activities were labeled correctly.

Bao et al. [1] used what they termed a "semi-naturalistic collection protocol" where participants completed an obstacle course and recorded their start and end times on a notepad with a pen. The course was designed to make participants perform certain activities that could be collected in a natural manner. To avoid ambiguity in the start and end of each task, they discarded 10 seconds of sensor data from the front and the back of the activities. While this method helps alleviate some of the burden on the experimenter, it is still cumbersome, has more ambiguous start and end markers, and would be obtrusive in a natural setting.

The experience sampling method (ESM) was used previously [17] to help automate the data collection task. In this use of ESM, data collectors went about their everyday lives, performing activities, and every 15 minutes, their device interrupted them to ask which of 35 potential activities they were performing. As the authors of this work point out, there are a number of problems with this approach. First, participants sometimes select the wrong activity. Second, due to the querying nature of ESM, there is limited granularity in the ground truth labels (15 minute chunks in this case). As such, short activities are difficult to capture with ESM and for long activities the exact start time is still unknown. Third, interruptions to the current activity can also be obtrusive, causing the data collector to inadvertently collect inaccurate data. Our VoiceLabel system avoids these problems by being less obtrusive and by maintaining user control of the collection and labeling process.

3. DESIGN CONSIDERATIONS

Labeling tools typically fall into three categories: offline, online, or a combination of the two. Offline labeling involves labeling sensor data after it has been collected. For example, a wearable camera (or camera in the environment) may be employed to record the video of what the participant was doing while sensor data was recorded. After the data is collected, a reviewer must examine the data and provide labels for activities. Depending on the complexity of the labeling, this can be slightly faster than real-time; more often, it is several times slower than real-time.

Online methods require the collector to indicate changes in activity in real-time as they engage in activities. Whenever the collector changes activity, they must indicate that transition through the data collection interface. Often the label from the online system will be cleaned up offline in a post-processing step to correct any errors. Because mobile devices are often used in online annotation, the limitations of the interface can be a source of error, (e.g., hitting the wrong button or delays caused by interaction with the device, and so forth).

In order to create better tools, it is important to understand what makes a good labeling tool. We present the following design considerations as a guideline for such properties. While not all of these issues can be solved in their entirety, we seek to design a tool that addresses the majority of our design considerations.

3.1 Size of Dataset and Speed of Labeling

Most supervised learning algorithms benefit from additional data. However, for activity recognition, collecting large datasets can be difficult, time consuming, and expensive. If the data collection requires a lot of dedicated time, collecting large data sets can be difficult. For example, Logan et al. [13] collected approximately 15 days worth of data, but due to budgetary constraints could only label 104 hours of data. In addition the monotony of the task can also lead to boredom, causing (even paid) labelers to become fatigued and generate poor data. Finally, properly segmenting streams of sensor data after collection can be challenging. For instance, it is often difficult to correctly align video to a sensor stream and manually find frames which correspond to the transitions due to poor camera angles, picture quality, ambiguity, and other issues. However, if data collection is easy, does not interfere with the activity, and can be integrated into the participant's normal schedule, then this can significantly ease the burden of collecting large data sets.

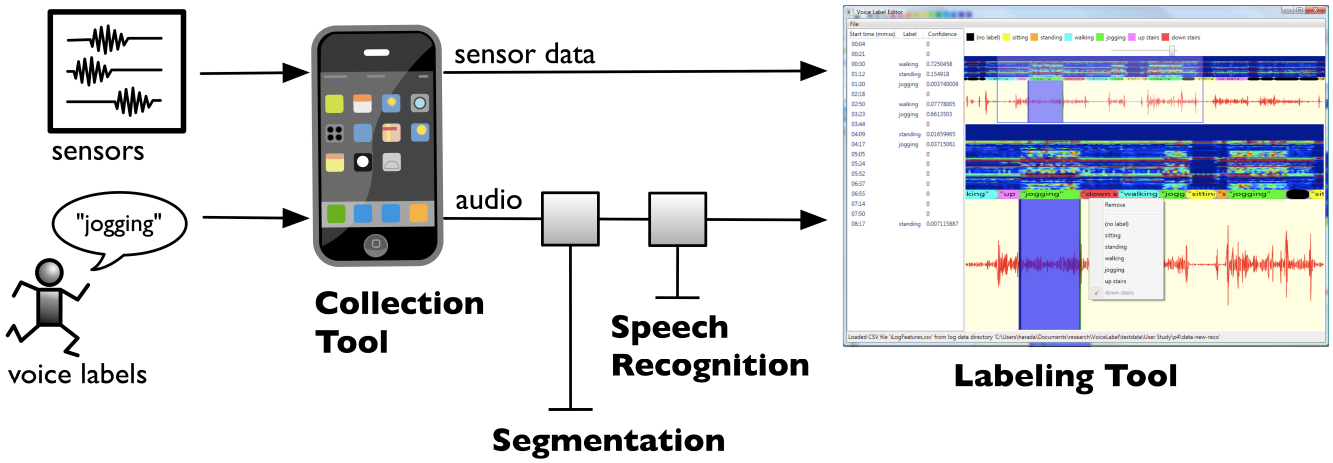

Figure 1: The VoiceLabel architecture. A mobile collection tool gathers sensor data and voice labels as the data collector engages in various activities. The collected data is processed through a segmentation module that identifies the activity transition points and a speech recognition module which attempts to extract the uttered label. The result can be viewed and manipulated via the labeling tool.

3.2 Unobtrusiveness

An ideal data collection and labeling system should be unobtrusive; the process of labeling the data should not affect the data being collected. But interacting with a GUI on a mobile device can be extremely obtrusive, especially when the device is physically attached to sensors like accelerometers.

Interaction and recording constraints may not always be physical in nature. Social constraints may also affect obtrusiveness of an interface. For example, recording audio or video might be considered acceptable when performing physical activities, but might be considered obtrusive if collecting data during a meeting or in public spaces where others may fear being recorded [14].

3.3 Verifiability

Poor segmentation and labeling can introduce noise (e.g., walking data might be mistakenly labeled as running, leading to a model which may poorly separate walking from running). To avoid such problems, it is important to verify the labeled data. Tools that allow multiple labelers to examine labeled data can help find inconsistencies and reduce errors. Correlation between labelers shows the inherent ambiguity of classification and, through outlier detection, weeds out poor labelers. In addition, labels that strongly conflict with classification can be re-examined to verify the accuracy of the original label.

As an online system will often lack adequate data sources to verify the labels offline, it is important for there to be relevant contextual cues gathered during the data collection phase so that labels can be corrected offline. For example, if we forego video recording of our data collection phase, it is important that we have enough contextual information to help decide what the actual label should have been. If the only data available is not humanly interpretable, it will be extremely difficult or impossible to determine what the actual activity was or to identify mislabeled segments.

4. VOICELABEL

To address these design considerations, we created an alternative to traditional labeling techniques for activity recognition. Our system, VoiceLabel, is composed of an online tool that allows data collectors to label data using speech, and an offline tool that allows labelers to review collected data.

4.1 Why Speech

We chose speech as the primary labeling modality for multiple reasons. Speech input removes the need for collectors to interact with a mobile GUI, allowing collectors to gather data more naturally. Although speech (and audio in general) may be inappropriate for some activities, a large number of activities, including the physical activities we are trying to model, benefit by removing the dependence on GUIs and moving towards speech.

Speech also allows labelers to verify the data after it has been collected. Labelers can listen to the speech label using the labeling tool. The speech label is a record of the activity while the activity was taking place, and data collectors are unlikely to misspeak and mislabel an activity. They are more likely to forget specific label names. For example, they might confuse “up stairs” with “walking up stairs.” This type of confusion is easy for human labelers to detect and correct offline. Addressing these errors can be as simple as making it easy for a labeler to identify unrecognized phrases, or more advanced support can be provided to add new phrases to the recognizer’s corpus.

Speech also reallocates the burden of data collection so it is easier for both data collectors and labelers. Collectors do some amount of work by speaking while they collect data, but this work is less than what they would have to do with a GUI on a mobile device and does not introduce spurious “signal” associated with manipulating the device. Because speech recognition is not perfect, labelers still need to spend some time verifying the results of the speech recognition. This work, however, is much less than would be required to manually annotate video. Both the labelers and the data collectors share some of the responsibility of gathering correct labels, and the distribution of labor leads to less work for both parties.

4.2 Architecture

VoiceLabel is composed of two separate components: the *collection tool* and the *labeling tool*, as shown in Figure 1. The

collection tool is a mobile application that collects sensor data and allows data collectors to label activities using speech. Data from the collection tool is then segmented and the audio is passed through a speech recognizer to get preliminary guesses at segments and labels. These guesses are then passed to the labeling tool, which allows labelers to verify and, if needed, correct the segments and labels.

When using the data collection tool, data collectors say the name of the new activity as they transition from one activity to another, marking an *activity transition*. The beginning of an activity necessarily implies the end of the previous activity, so a move from walking to standing is noted by just saying "standing."

To split data into *activity segments* and associated *activity labels*, VoiceLabel needs to detect transitions between activities. For example, VoiceLabel needs to know at what moment a person changed from walking to standing. All the data before the transition should be labeled as walking and all the data after the transition should be labeled as standing. Transitions can be detected offline by running a speech recognizer over the entire audio stream and looking for the names of activities. However, low accuracy in the speech recognizer may lead to poor segmentation. It is difficult to recognize when an activity transition took place. Explicitly indicating a transition online during data collection (e.g., with a button press) helps the system segment activities.

5. IPAQ PROTOTYPE SYSTEM

To understand the feasibility of using speech as the method for assigning labels during data collection, we created an initial prototype of the collection tool that used the push to talk (PTT) method for indicating the activity transition and marking the beginnings of each label name utterance.

5.1 System Description

Our initial prototype of the collection tool ran on an iPAQ hx7400 PDA connected via USB to the Mobile Sensing Board (MSB). The MSB did all of the sensing, including audio, which was used to label the sensor data. To clearly pick up audio, the MSB was attached to the shirt collar or lapel.

Since all sensing occurred on a device separate from the PDA, physically manipulating the PDA did not affect the sensor data. As a result, we used the physical button on the PDA to indicate transitions between activities. On a transition, the data collector pushes the button and says the activity. Since this interface did not provide GUI feedback, we played a small chime when the data collector pressed the physical button to indicate that they had successfully initiated a transition.

We also built a prototype labeling tool that showed each segment along with the recognized activity and confidence. Labelers could listen to the audio for each segment to verify the output of the speech recognition system. Incorrect labels could be changed by selecting the correct activity from a drop down menu.

5.2 Initial Feasibility Study and Results

We conducted an initial feasibility study to explore people's ability to label activities on a mobile device using their voice. Four participants were asked to collect data for the following eight activities: (1) sitting, (2) standing, (3) walking, (4) jogging, (5) going up stairs, (6) going down stairs, (7) going up an elevator, and (8) going down an elevator. The experimenter reviewed the activities with the participant, instructed the participant on how to label data, and led them along a course on which they performed each of the activities at least once. Participants were not explicitly told when to label a new activity; they had to remember to label transitions.

Participants labeled data using both voice and GUI collection tools. When using the voice collection tool, participants indicated activity transitions by pressing a button on the mobile device and saying the name of the activity. The GUI collection tool presented a list of activities with checkboxes next to them which can be selected using a stylus. In our study, participants were asked to collect data using both interfaces and then to comment on them.

Interacting with the mobile device can affect a person's physical activity. For example, participants may stop moving when trying to select their current activity from a screen. Experimenters noted the amount of time participants spent interacting with the mobile device. Our results showed that participants spent less time interacting with the mobile device when using our prototype than they did when using the GUI.

When asked which tool they would prefer to use, three out of four participants stated that they preferred the VoiceLabel prototype to the GUI. Participants felt that it was easier to annotate using VoiceLabel because they did not have to stop and interact with the GUI on the mobile device. The final participant preferred the GUI because he felt the GUI helped him remember the set of activities. This is a key limitation in labeling by voice. Remembering the labels for all eight activities can be difficult, especially for first-time users.

6. VOICELABEL ON THE IPHONE

Based on the knowledge we gained from the initial prototype, we created a new version of the system in which the collection tool ran on an iPhone and the labeling tool included a much richer set of features for reviewing and correcting the automatically extracted labels. We call this version of the system iVoiceLabel and provide a description of its features below.

6.1 iVoiceLabel System

6.1.1 Collection Tool

The collection tool is written in Objective-C and compiled for the iPhone OS 1.1.4. After the tool is started, it begins recording sensor readings from the onboard accelerometer as well as capturing audio from the headset microphone, logging both to file. When the user is finished with the series of activities, the saved data is uploaded to a computer running the labeling tool.

One key difference in the collection tool for iVoiceLabel, as compared to the initial prototype, is the way in which an activity transition is indicated. Whereas our initial prototype required a button press before uttering the label name, the iVoiceLabel system simply requires the user to speak into the headset microphone, preceding the label name utterance with a "filled pause." A filled pause is a form of vocalization in which a particular vowel sound is held for some duration (e.g., "uhhhh"), and has been used as a voice input technique in a number of scenarios [8]. During this recording process, the user does not have to physically interact with the iPhone.

Using a filled pause as an indicator for the start of an activity segment has some advantages. Unlike our prototype system, in which button presses were used to indicate transitions, filled pauses allow the entire data collection process to be hands-free. This hands-free setup reduces the chance of interrupting the

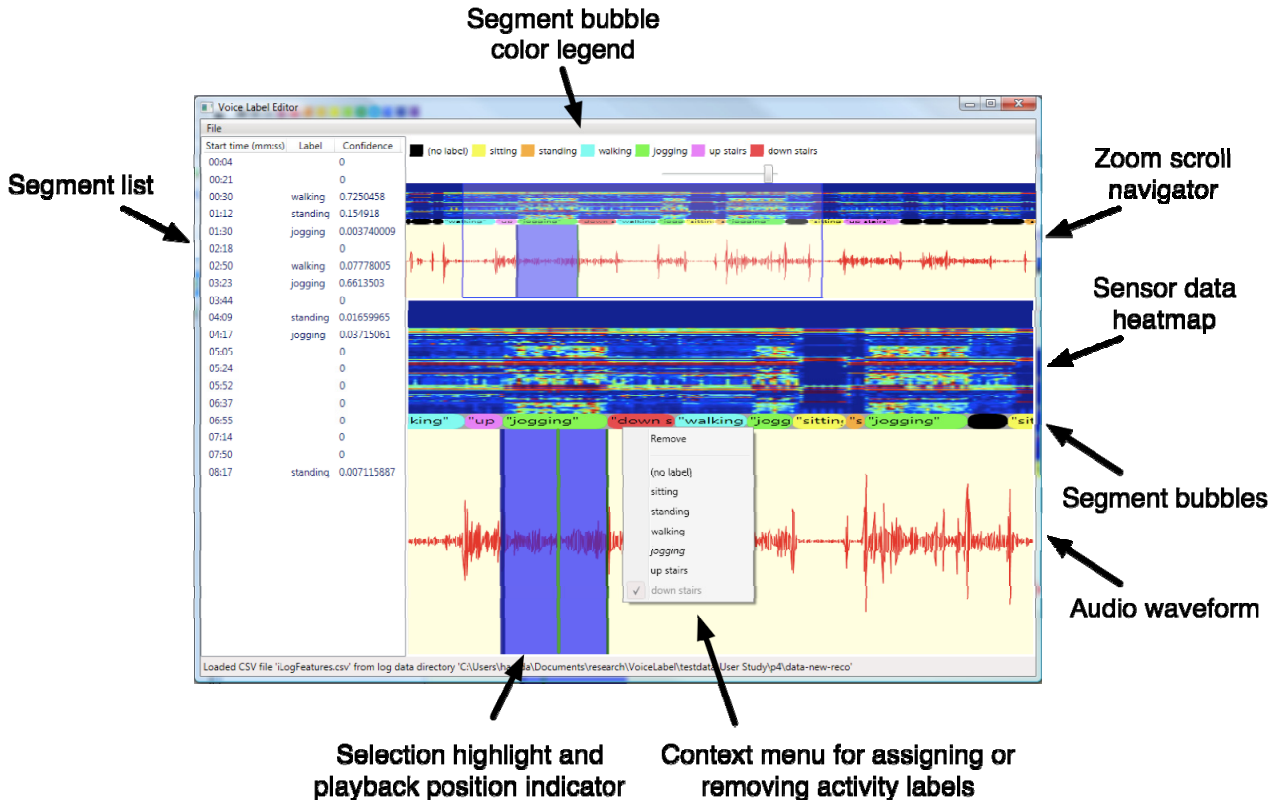

Figure 2: Screenshot of the VoiceLabel labeling tool. The tool allows the labeler to easily browse through the sensor data recorded during the data collection phase, along with the audio waveform containing the activity labels. It also automatically extracts activity transition points as segment bubbles containing the recognized activity label; these automatic labels can be corrected if, after listening to the corresponding audio, they are determined to be incorrect.

activity and enables data collection even in hands-busy situations such as driving or biking. However, this advantage comes with tradeoffs. First, a button-push is an unambiguous signal to the system that data should be segmented. While filled pauses can be detected relatively reliably by tools such as the Vocal Joystick engine [3], filled pause recognition is not perfect. The segmentation and subsequent tools therefore need to take into account both recognition accuracy for spoken labels and filled pauses. Second, a human labeler may not be able to indicate the exact moment of transition. To account for this, many activity recognition systems discard data surrounding the transitions.

6.1.2 Labeling Tool

When the labeling tool is started, it looks for the data uploaded from the iPhone and begins the processing phase to automatically extract as much labeling information as possible based on the verbal utterances. The tool first passes on the recorded audio file to the filled pause segmentation module.

The filled pause segmentation module uses the Vocal Joystick engine to detect when one of the vowel sounds is sustained for at least 400ms (empirically determined to be representative of natural filled pause utterances). For each detected filled pause, the labeling tool passes on the next three seconds of audio to the speech recognition module.

The speech recognition module uses the SAPI 5.3 speech recognition engine on Windows Vista to recognize the label that was uttered following a filled pause. A simple fixed vocabulary grammar-based language model is used in which the only possible utterances are the valid label names. The results from the recognition along with the timestamp corresponding to the filled pause starting points are then displayed on the GUI interface along with the visual representation of the sensor data and the audio waveform.

Figure 2 shows the interface for the labeling tool in iVoiceLabel. On the left is a list of all the filled pause segments detected by the filled pause segmentation module along with their corresponding recognition results. On the top-right is an overview of the entire data stream with a current focus area illustrated by a translucent blue highlight. Details of the focus area are displayed on the bottom-right. The data stream is rendered as a stack of three graphs. The top graph with the darker bluish background is the heatmap showing the values from the sensor data, where each row of pixels corresponds to some particular sensor value (either raw or processed). In our case, we display 125 features that contain both the raw accelerometer data as well as calculated features such as the FFT coefficients for each axis. The bottom graph with the lighter yellow background is the waveform graph of the recorded audio, aligned to the sensor data heatmap. Between these two graphs is a thin band of "segment bubbles," which are color-coded rectangular regions corresponding to the automatically extracted activity segments. The corresponding activity label names are shown in the bubbles as well as in the legend at the top of the screen.

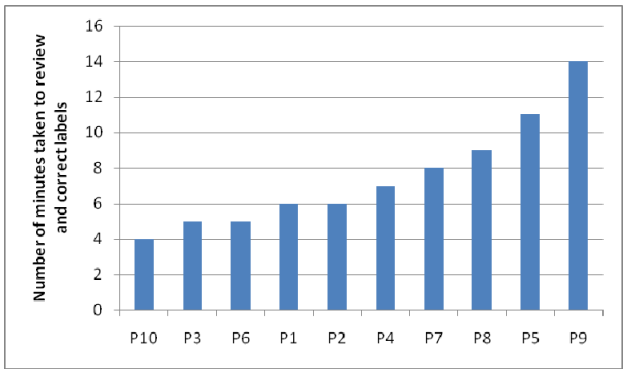

Figure 3: Time it took each participant to complete the label correction task on the labeling tool.

The labeling tool supports a number of interactions for reviewing and modifying the automatically extracted activity segments and their labels. First, the mouse wheel can be used to scroll into any portion of the data stream, and the preview pane supports panning. The user can also click on any of the segment bubbles to initiate playback of the corresponding audio stream. When a segment is clicked, the region on the waveform corresponding to that segment is highlighted in blue and a vertical green line representing the current playback position begins to animate to show where in the audio stream is being played. As the user listens to the audio, if they detect that the actual uttered label is different from what has been automatically extracted, they can right click on the segment bubble to assign the correct label using the context menu. If the detected segment was spurious (i.e., there was no actual filled pause followed by a label name), they can remove it through the context menu as well. They may also click anywhere on the waveform to select a region from that point on to the end of the corresponding segment and initiate playback. This is crucial in cases where the filled pause segmentation module failed to identify an activity transition, but the user can see from the waveform and/or the heatmap that there was something uttered or that something changed in the sensor reading. If the selected region does indeed contain an actual label utterance, a new segment can be created at that selection by right clicking on the highlighted region and selecting the corresponding label. Using these interactions, the user can review and edit the activity segments to match what actually happened.

6.2 Laboratory Evaluation

To better understand the strengths and limitations of iVoiceLabel, we conducted a laboratory evaluation in which participants used the tool to collect and label sensor data. We recruited 10 participants (6 male, 4 female). Each participant was asked to play the role of both the data collector and the labeler. The study took between 30 to 45 minutes per participant.

Participants were first given an overview of the experiment. They were told that they would be labeling six distinct activities (sitting, standing, walking, jogging, going up stairs, and going down stairs) using their voice. The participant was then given the iPhone and led through the building by an experimenter. The experimenter directed the participant along a preset course by telling them what activities they should be doing (e.g., "ok now walk down the hall and go up to the third floor"). The course consisted of 18 activities (three each of the six activity types), and traversing the course took approximately 10 minutes.

After the participant finished performing the activities, the data was transferred to a Windows Vista machine with an external 24 inch monitor and processed. The participant was then presented with the labeling tool, given a brief tutorial in which they were able to try out the interface on small data set, and then asked to review and correct their own data until they felt it represented what they had done. The labeling task, including the tutorial, took about 20 minutes. Participants were encouraged to think aloud during the labeling task. After completing the labeling task, participants completed a short questionnaire and discussed their answers with the experimenter.

7. RESULTS AND DISCUSSION

7.1 Labeling Results

Figure 3 shows the amount of time the participants took to complete the review and correction process. Most people took substantially less time than the actual length of the activity collection (i.e., they did not have to sequentially listen to the entire audio stream). There were a few participants who spent almost as long as or longer than the activity duration. These participants tended to take multiple passes through the sensor data, involving at least one pass where they listened through almost the entire audio stream.

The overall precision of the filled pause detector was 86.7% (stdev: 11%), with recall of 65.2% (stdev: 20.5%). Some of the factors that seemed to throw off the filled pause detector were elevator chimes (which sound like a sustained tone) and the variability in different individual's vowel utterances (we used a speaker-independent acoustic model for the filled pause detector). We believe that further tuning is possible to decrease the false positives and increase the accuracy rate by adapting the filled pause detector to each individual participant's voice. Also, even in cases where the segmentation module failed to detect the filled pause because the microphone was fairly close to the person's mouth, the audio waveform clearly showed peaks for when they made the utterance.

After participants used the labeling tool to review and correct the segments and their corresponding labels, all but two users managed to identify all of the activity transitions and label them with correct activity labels. The two who did not manage to do so only missed one activity transition each. In both cases, the transition was not prominent in both the audio waveform (there

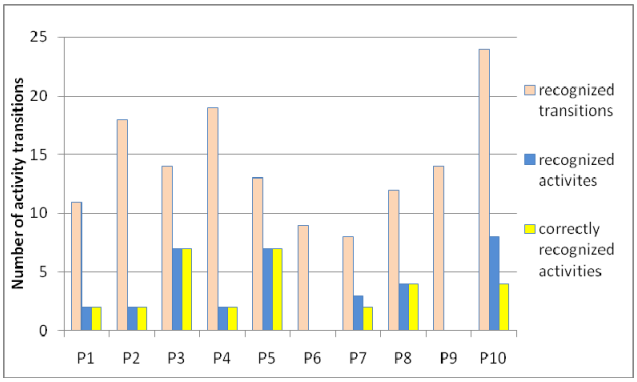

Figure 4: Number of activity transitions for each participant that were: (a) detected by the segmentation module, (b) recognized to be some activity by the recognizer, and (c) recognized to be the correct activity by the recognizer.

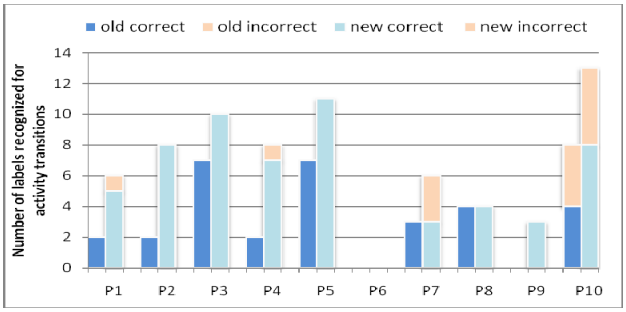

Figure 5: For each participant, the number of correct and incorrect labels recognized under the old recognizer (left) and the new recognizer (right).

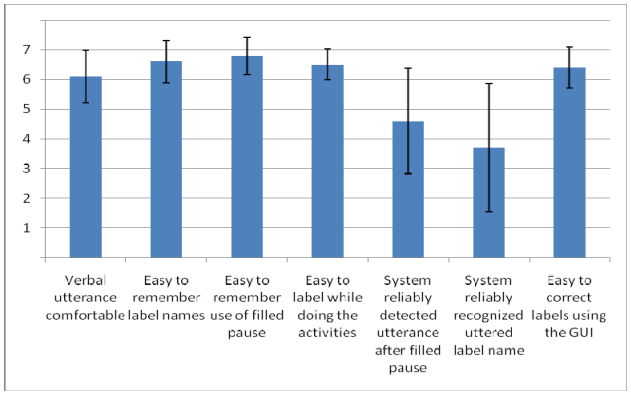

Figure 6: Results from the post-study 7-point Likert-scale questionnaire (1 = strongly disagree, 7 = strongly agree).

was too much noise) and the sensor data heatmap (one of the transitions was from going up stairs to down stairs).

The initial recognition rate from the speech recognizer was not very encouraging, with an average of only 26% of the valid transition utterances being recognized. After seeing how low the recognition rates were, we went back to try to see if we could boost the accuracy rate by relaxing the constraints on the recognizer. Instead of only considering the successful recognition results as reported by the speech recognizer, we extracted all the intermediate hypothesized results along with their confidence values. Even when there was no successful recognition event reported, we took the hypothesis with the highest confidence value as our result. We conjectured that due to the relatively small number of label names, the failed hypotheses may still have a good chance of being correct and that users would benefit more from being presented an incorrect hypothesis than no hypothesis at all. Figure 5 shows the comparison of the results obtained in this fashion to those obtained in our original design. The new results show a marked increase in the total number of recognized segments as well as the number of correctly recognized segments.

7.2 Qualitative Feedback

The results of a follow-up questionnaire are shown in Figure 6. Overall, all participants rated all but one category favorably (perceived quality of recognition results), expressing their comfort level with using audio to label activities and the ease of use of the labeling tool. Regarding the social comfort level of speaking the label names while performing the activity, participant P3 stated, "I see people talking to themselves on their Bluetooth headset all the time, so I don't think it's particularly awkward."

The ratings for their perceived reliability of the segmentation and recognition modules were bimodal. This bimodal nature is also reflected in the actual result as shown in Figure 4. This seems to stem from the nature of the participant's speech, where certain participant's voices were much quieter or faster than others, throwing off the segmentation and recognition modules.

Most participants found the combination of sensor data heatmap and waveform helpful. For example, P7 commented, "the audio data alone wasn't enough to pick out portions of the data stream that might contain an activity transition, particularly with ambient noise," and stated that the combined display of the sensor data and the audio waveform "made it easier to have confidence [that he has identified all of the transitions] without listening to the entire stream." P3 also utilized the heatmap to notice that certain activities have a distinctive color pattern, and therefore he was able to save time by skiping over segments with the same pattern. Those who mentioned that they did not use the heatmap explained that their audio waveform was distinctive enough and the peaks corresponding to their filled pause utterances were easy to pick out. This points to the powerful effect of having multiple sensor modalities available at the time of review to supplement each other in case one is ambiguous or noisy. Eight of the participants explicitly commented that they really liked the labeling tool and that they found it easy and fun to use to quickly locate activity transitions and choose a label.

7.3 Observations

The results from our laboratory evaluation suggest that the use of speech input is a viable form of real-time mobile sensor data collection and labeling and that a multimodal labeling tool that combines graphical sensor data representation and audio output can greatly facilitate the data labeling and correction process. We gathered the following observations:

- The use of filled pauses is a reasonable method for marking activity transitions when there is little ambient noise. Even when the filled pause detection does not work well, the waveform associated with a filled pause is easily spotted by a labeler using our labeling tool.
- The combined presentation of the sensor data and the audio data in a time-aligned fashion provides rich contextual cues, where each data source can help disambiguate certain portions of the data stream in which there are no salient features in the other data source. This allows the quality of each data source to be less than perfect yet still provide enough cues to the user.

8. FUTURE WORK

VoiceLabel allows users to collect and label activity data in an easier and less obtrusive manner, but the use of voice has its own limitations. Certain activities are limited by physical and social constraints: speech may be inaudible in noisy environments and inappropriate in a meeting. Future work could examine different input modalities for labeling based on the collector's constraints. For example, a collector might prefer to use a GUI interface when gathering activities in a meeting and a speech-based interface when driving to work. Different labeling modalities come with different tradeoffs, and our labeling tool would need to be modified to help labelers understand how labels were collected.

In our current system, the majority of speech processing happens offline. However, current commodity hardware has enough computational power to perform some of this processing online. Online processing has a number of benefits, most notably allowing a system to provide immediate feedback. For example, when a filled pause is recognized, the system can chime and wait for the activity name to be uttered. After the utterance is made, the system can repeat the recognized activity to the collector to verify the result. This can greatly reduce the offline effort by the labeler while maintaining a completely hands free interface.

Our current segmentation and speech recognition modules use general speaker models. By adapting these models to each individual data collector, we can provide the labeler with much more accurate segmentation and recognition results. One way in which such models can be adapted is to create a mixed initiative system that adapts the model as the labeler corrects or verifies the recognition results [15]. For example, if a labeler creates a new segment or verifies a recognized label, the system can modify its model and update its guesses for data the labeler has not reviewed.

By leveraging existing classifiers or unsupervised methods, VoiceLabel can provide better segmentation and recognition. Unsupervised methods can be used to cluster similar sensor data. Data from different activities should fall into different clusters, and indicating different clusters in the offline tool might help the labeler find segments they would have otherwise missed. Additionally, if a labeler verifies a label for one segment, the label might propagate to other segments in the same cluster. This information can provide richer information to the labeler, increasing their speed and accuracy. Existing classifiers can be used in a similar way to unsupervised methods and provide additional information in the form of label names.

9. CONCLUSION

Labeling mobile sensor data for building activity recognition applications poses a number of challenges. The process of labeling the activities must be fast, not taint the sensor data, and retain enough contextual cues to allow for offline validation.

Voice labeling is an effective method for collecting ground truth data for building activity inference models on mobile devices. Our initial prototype showed that a push-to-talk system provides perfect segmentation; however, our final filled pause system was more intuitive and less obtrusive. Our offline tool for labeling was intuitive and easy to use by participants. It enabled them to quickly and effectively correct segmentation errors and apply the correct label to segments. We have demonstrated the effectiveness of filled pause annotation in enabling researchers to collect natural data with little effort. This work will help further the development of high-quality sensor-driven applications for mobile computing.

10. ACKNOWLEDGMENTS

We thank our study participants for their time and feedback. This work was supported in part by the National Science Foundation under grant IIS-0326382, by Kayur Patel's NDSEG fellowship, by Intel Research and by Microsoft Research.

11. REFERENCES

[1] Bao, L. and Intille, S. S. 2004. Activity recognition from user-annotated acceleration data. In Proc. of Pervasive 2004.

[2] Basu, S., Clarkson, B., and Pentland, A. 2001. Smart headphones: enhancing auditory awareness through robust speech detection and source localization. In Proc. of ICASSP 2001.

[3] Bilmes, J., Li, X., Malkin, J., Kilanski, K., Wright, R., Kirchhoff, K., Subramanya, A., Harada, S., Landay, J., Dowden, P., Chizeck, H. 2005. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In Proc. of HLT/EMNLP 2005.

[4] Chen, M, Sohn, T., Chmelev, D., Haehnel, D., Hightower, J., Hughes, J., and LaMarca, A. 2006. Practical metropolitan-scale positioning for GSM phones. In Proc. of Ubicomp 2006.

[5] Chen, S., Gaur, A., Muthukrishnan, S., and Rosenbluth, D. 2004. Wireless in loco sensor data collection and applications. In Proc. of WWW 2004 Workshop on Emerging Applications for Wireless and Mobile Access.

[6] Consolvo, S., McDonald, D., Toscos, T., Chen, M., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., and Landay, J. 2008. Activity sensing in the wild: a field trial of UbiFit garden. In Proc. of CHI 2008.

[7] Fontaine, V., and Bourlard, H. 1997. Speaker-dependent speech recognition based on phone-like units models – application to voice dialing. In Proc. of ICASSP 1997.

[8] Goto, M., Itou, K., and Kobayashi, T. 2005. Speech interface exploiting intentionally-controlled nonverbal speech information. In Proc. of UIST 2005.

[9] Hill, J., Szewczyk, R., Woo, A., Hollar, S., and Pister, D.C.K. 2000. System architecture directions for networked sensors. In Proc. of ASPLOS 2000.

[10] Lamming, M., and Bohm, D. 2003. SPECs: Another approach to human context and activity sensing research, using tiny peer-to-peer wireless computers. In Proc. of Ubicomp 2003.

[11] Lester, J., Choudhury, T., Kern, N., Borriello, G., and Hannaford, B. 2005. A hybrid discriminative-generative approach for modeling human activities. In Proc. of IJCAI 2005.

[12] Lester, J., Choudhury, T., and Borriello, G. 2006. A practical approach to recognizing physical activities. In Proc. of Pervasive 2006.

[13] Logan, B., Healey, J., Philipose, M., Tapia, E.M., and Intille, S. 2007. A long-term evaluation of sensing modalities for activity recognition. In Proc. of Ubicomp 2007.

[14] Nissenbaum, H. 1998. Protecting privacy in an information age: The problem of privacy in public. In Law and Philosophy 17 (5-6).

[15] Shilman, M., Tan, D., and Simard, P., 2006. CueTIP: A mixed-initiative interface for correcting handwriting errors. In Proc. of UIST 2006.

[16] Saponas, T., Lester, J., Froehlich, J, Fogarty, J., Landay, J. 2008. iLearn on the iPhone: Real-Time Human Activity Classification on Commodity Mobile Phones. University of Washington CSE Tech Report UW-CSE-08-04-02.

[17] Tapia, E.M., Intille, S.S., and Larson, K. 2004. Activity recognition in the home setting using simple and ubiquitous sensors. In Proc. of Pervasive 2004.