# PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones

Kevin A. Li, Timothy Sohn, Steven Huang, William G. Griswold
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
{k2li, tsohn, sbhuang, wgg} @ cs.ucsd.edu

## ABSTRACT

Mobile phones have the potential to be useful agents for their owners by detecting and reporting situations that are of interest. Several challenges emerge in the case of detecting and reporting "nice to know" situations. Being alerted of these events may not be of critical importance but may be useful if the user is not busy. For detection, the precision of sensing must be high enough to minimize annoying false notifications, despite the constraints imposed by the inaccuracy of commodity sensors and the limited battery power available on mobile phones. For reporting, the notifications cannot be too obtrusive to the user or those in the vicinity. Peripheral cues are appropriate for conveying information like proximity, but have been studied primarily in settings like offices where sensors and cueing mechanisms can be controlled.

We explore these issues through the design of PeopleTones, a buddy proximity application for mobile phones. We contribute (1) an algorithm for detecting proximity, (2) techniques for reducing sensor noise and power consumption, and (3) a method for generating peripheral cues. Empirical measurements demonstrate the precision and recall characteristics of our proximity algorithm. A two-week study of three groups of friends using PeopleTones shows that our techniques were effective, enabling the study of how people respond to peripheral cues in the wild. Our qualitative findings underscore the importance of cue selection and personal control for peripheral cues.

## Categories and Subject Descriptors

H.1.2 [Information Systems] User/Machine Systems – Human Factors; H.5.2 [User Interfaces] Evaluation/Methodology, User-Centered Design

## General Terms

Human Factors

## Keywords

Context-Aware Systems, Peripheral Cues, Mobile Computing, Ubiquitous Computing, Proximity Detection


*MobiSys'08*, June 17-20, 2008, Breckenridge, Colorado, USA.


## 1. INTRODUCTION

One vision for ubiquitous computing is a context-aware infrastructure that can simplify and enrich our lives by helping us with tasks that might otherwise be out of our reach. For example, location-based services such as Loopt can detect the proximity of friends that are just out of sight or unnoticed [27]. Such applications can be useful for a variety of scenarios such as arranging ad-hoc meetings. To date such wide-scale applications have depended on specialized phone and carrier capabilities to detect proximity, both at a real cost to the user. Moreover, the user must make a conscious effort to look at the phone to learn of friends' proximity, lessening usefulness.

Realizing the ultimate vision depends on a ubiquitous mechanism for detecting such occurrences. For "nice to know" contextual information like the proximity of friends, we also need an unobtrusive mechanism for making us aware of them. To achieve true ubiquity – so that any two friends could be aware of their proximity – both must be achieved at little cost. In this paper we explore the technologies of mobile phones and peripheral cues for the ubiquitous sensing and reporting of "nice to know" context through PeopleTones, an application for buddy proximity:

- Commodity mobile phones satisfy the ubiquity criterion (and by extension the cost criterion). As of 2007, there are 3.3 billion mobile phone subscribers worldwide [45]. Moreover, mobile phones possess both a number of sensors (e.g., microphone, camera, and GSM radio) and actuators (e.g., speaker and vibration motor), making phones a potentially ideal platform for ubiquitous computing. On the other hand, the sensors and actuators are of notoriously low quality, complicating precise sensing and high-fidelity actuation. Inference can be especially problematic when comparing readings between phones [11].
- Peripheral cues, like those explored in office and home environments, are an attractive modality for "nice to know" information; they can apprise users of information without interrupting their current task. However, getting peripheral cues to work with commodity mobile phone actuators in the wild is an open challenge.

For detecting proximity on phones, our algorithm compares cell towers seen by the mobile phone clients to estimate proximity. This privacy-friendly approach does not require knowledge of actual location. However, GSM's long range and random characteristics means that a phone will, for example, occasionally detect cell towers that are miles away. We filter the proximity data using a simple state machine based on a 2-bit counter [33]. The state machine also helps to conserve power by sampling less frequently when two phones are considered near or far away. Power is further conserved by withholding reports when GSM signals are weak; proximity detection in this case is imprecise and extra power is required to report it.

Proximity can be reported by sounds, and past work has shown audio to be effective for delivering peripheral cues [32]. However, it is untenable to expect the use of headphones or similar devices to reduce the unobtrusiveness of cues or increase comprehension. We hypothesize that keeping audible cues short can improve their unobtrusiveness, but that may not be adequate for many uses. We propose using the vibrations provided by mobile phones, as they are private, subtle cues [17], and likely to be etiquette-friendly. Vibrotactile cues that correspond to known audible cues (i.e., they "vibrate like the sound") can provide a parallel "private" vibration language without requiring the user to learn an arbitrary mapping.

However, the inexpensive vibrotactile actuators found on mobile phones today only have a binary on/off setting, severely limiting their communication abilities. To provide vibrotactile cues corresponding to the audio cues, we introduce an offline digital signal processing technique that captures the essence of audio cues. These patterns were realized on the mobile phone's limited vibrotactile actuator using our software algorithm. Using a technique similar to pulse width modulation, we can generate a range of amplitudes.

To explore these mechanisms we performed both controlled and *in situ* user studies of PeopleTones. First, we measured the precision and recall of our proximity detection algorithm using a large dataset collected from wardriving the Seattle area [11]. Second, we lab-tested 17 users on their ability to identify how vibrations corresponded to music clips. Finally, we designed and deployed PeopleTones, a system for conveying buddy proximity via peripheral cues that are uniquely assigned to buddies. PeopleTones was deployed to three groups of friends (the same 17 users as above). Each group used a different cue-to-buddy mappings: nature sounds, music sounds picked by the buddies, and music sounds picked by the recipient of the cues. To uncover possible learning of the vibrotactile cues during the study, we repeated the lab test at the completion of the study.

For proximity detection, our findings show that our approach has excellent precision (few false positives) and fair recall (a split between true positives and false negatives). The two bit counter reduces false positives by up to 84.9% and increases precision to 99% at a threshold ratio of 0.4. The fair recall is adequate because proximity is most valuable for people who are lingering near each other (e.g., not driving), and such behavior provides many chances to produce a positive report. Such lingering also diminishes the effects of the cell towers that were cached in the phone upon arrival to the area, which is influenced by the towers seen along the users path to the destination. Managing power by dynamically adjusting the sampling rate enables a phone to run for one to two days, as opposed to 4-6 hours.

A general theme of the qualitative results is the importance of personal control for peripheral cues. Using ambient noise to detect social situations was explored as a way of choosing audio versus vibration cues, but most users opted to enforce explicit control of their cue delivery modality. Users who selected the cues they would hear found the system more useful and were also better at identifying corresponding vibrotactile patterns generated by our algorithms.

After a discussion of related work, Sections 3 and 4 detail our approaches to proximity detection and delivering peripheral cues on mobile phones. Section 5 introduces PeopleTones, and Sections 6-9 describe the results of our *in situ* user study.

# 2. RELATED WORK

We build on work from three areas: proximity sensing, mobile peripheral cue systems, and auditory and tactile cues.

## 2.1 Proximity Detection in the Wild

A number of technologies have been proposed for proximity sensing. Infrared approaches such as those used by *Meme Tags* [3] provide good accuracy but require line of sight. Ultrasound approaches such as *Activebadge* [42] also provide good accuracy but they require infrastructure support. *Hummingbird* uses short range radio which allowed Holmquist et al. to explore deployments in the wild [20]. This approach provided good proximity detection but required specialized hardware, which created complications for in-the-wild deployment. *PlaceLab* uses estimates of cell tower positions to provide location [26]. This approach provides excellent coverage and adequate accuracy for detecting something like buddy proximity (e.g., median accuracy of 94-196m and $90^{th}$ percentile accuracy of 291-552m, using a single carrier's towers), but it requires a "wardriving" of the area to obtain location estimates for cell towers in the area. This can be quite costly, especially keeping the information up to date, as tower positions, etc. are updated on an annual basis.

One method of acquiring location on some phones is through the network carrier, but they often do not release the required APIs. *Loopt* is an example of a commercial system that enables sharing location with friends using both GPS and carrier based location [27]. Unfortunately GPS is not yet widespread, suffers from not working everywhere, i.e. urban canyons and would violate user's privacy by requiring location reports to a server. *Dodgeball* explores self-reporting location [12] but would require user's to proactively monitor the system, which is counter to our goal.

Another method is to use a location infrastructure. *Place Lab* [26], *Active Campus* [16], and *Plazes* [34] are examples that offer both absolute and relative positioning. These infrastructures limit sensing to areas with pre-mapped access points.

Rather than calculate absolute location, NearMe explores a few algorithms for detecting proximity using Wi-Fi signatures, allowing it to work with no *a priori* setup [25]. We use a similar approach but for GSM readings.

## 2.2 Mobile Peripheral Cue Systems

Peripheral cues have been heavily examined in office settings. *Audio Aura* [32], *Live Wire* [43] and *ambientROOM* [24] are systems that play auditory cues for conveying information in the background. Peripheral displays are known to be difficult to evaluate [29,31] and peripheral cues suffer from similar problems. The success of peripheral cues in home and office environments suggests that they may be useful in the wild. Deployments in the wild often reveal uses not found in laboratory studies. Examples of this include location-sharing [38] and reminders [39].

Mobile context-aware platforms have been proposed for aiding instant messaging, an intended, explicit interaction scenario.

*WatchMe* [30], *Hubbub* [22], and *Connexus* [41] are examples of such applications, supporting the initiation of a messaging session by providing cues of availability (e.g., not in a conversation). Studies with *Nomadic Radio* found that auditory communication is useful for mobile messaging but minimizing intrusiveness requires more than, for example, detecting breaks in conversation [36]. One possibility is detecting activity transitions with accelerometers placed in the seat of a chair or worn on the body [19]. Such approaches are less viable in the wild. Many of these systems have suggested that audio cues could be used to identify different users. Commercial ringtones are similar in that they map a person's identity to an audio cue, but they are little studied and most phone users don't consider an incoming call a "nice to know" condition (and hence worthy of a peripheral cue).

## 2.3 Auditory and Tactile Cues

Gaver's work with auditory icons revealed the effectiveness of using sounds that are semantically related to the objects they represent [13]. Brewster's work with earcons found that music timbres are better at conveying information than unstructured sounds [5] and that non-speech audio can be effective for navigation [4]. We build off of these findings, using structured sounds for our auditory cues and exploring how different types of sounds effect user response.

Tactile cues are subtle, private cues [17] that have been suggested as a channel for ambient information delivery [35]. Tactile perception cannot be fully utilized without a high-fidelity delivery channel [1]. Piezoelectrics have been proposed to convey information using touch, such as in Luk et. al's *Tactile Handheld Miniature Bimodal* [28]. Vibrotactile cues have been proposed for a variety of uses such as for conveying information in a non-visual channel. Geldard's *Vibratese* language proposed a vibrotactile encoding of the English alphabet [14,15] and *ComTouch* explored vibrotactile communication without learning (i.e., training) [9]. *Tactons* use specialized actuators similar to those found in mobile phones to generate distinct pulses, which have been shown to be effective for alerting users to message type as well as urgency [7,8]. These works demonstrated that vibrotactile patterns can be differentiated. Multifunction transducers have been used to explore audio-haptics, playing vibration in conjunction with audio in mobile phones [10]. Still, vibrotactile development on commodity phones is limited by APIs that provide only on/off functionality. The *VibeTonz* technology from Immersion supports richer, more complex vibrotactile pattern generation, but utilizes specialized hardware that is currently available on only a handful of commercially available handsets [23].

# 3. PROXIMITY DETECTION

There were two design requirements we felt were necessary for a buddy proximity detection algorithm. First, it should be widely deployable in many environments with many phones, doing so in a privacy-aware manner. Secondly, since buddy proximity is "nice to know" information, it is important that when cues are delivered, friends are actually near one another. If too many cues will be delivered when buddies are far away, users will stop using it. In the case of reporting when buddies are near, it is therefore important to maintain a high *precision*, even if this means lower *recall*.

*Precision* is defined as the number of near reports that are correct divided by the total number of near reports. High precision means that there are few false positives. *Recall* is the number of near reports that are correct divided by the total number of actual near occurrences. High recall means that most of the near occurrences have been detected.

PeopleTones does not need a person's geographic location to find the proximity of nearby buddies. Hence, we used a relative positioning method in the spirit of the Nearme server [25]. Nearme used a variety of metrics for comparing the distance between two wireless measurements, such as Euclidean distance, spearman rank correlation, and the ratio of common access points.

## 3.1 Proximity Detection Algorithm

To run controlled tests on a few different proximity detection approaches, we collected a small sample of cell tower readings from three regions with different population densities. These were obtained by sampling cell tower information from each of 3 mobile phones, all on the same carrier. Each phone recorded two samples while positioned each location. We took samples 5 minutes apart to approximate realistic behavior where users might linger at a particular location. To eliminate potential caching effects that may occur when reading cell tower information from the phone's memory, we reset all the phones in-between samples. One phone was kept stationary while the other two were moved away from the stationary one at 0.2mi intervals. The i-mate SP3i (HTC Tornado) phones we used are capable of reporting up to 7 towers at once. In summary, we used 2 samples per *phone* per *region*, 2 *phones*, 7 *distances*, and 3 *regions*, resulting in 84 readings. The purpose of gathering these readings was to test different algorithms for proximity detection on a realistic set of data. In our initial experiments, we found that computing the ratio of common GSM cell towers between two readings provided the best real-time proximity indicator. The intuition is that the closer two phones are, the more cell towers they will have in common. This ratio is simply the number of common towers between the two phones divided by the average number of towers seen. Figure 1 shows the equation we used for computing the ratio of cell towers between two phones, given two readings *a* and *b*, each consisting of a set of cell tower sector identifiers.

Figure 2 plots the averages of the *proximity-ratio* values for the three regions from which we collected data. From this plot we can see a clear trend for ratios to decrease as distance increases, although not consistently; there is a lot of noise. This suggested that the ratio approach would be promising for approximating distance between two stationary phones but we still needed to determine an appropriate ratio for a peripheral cue application's needs.

$$proximity\_ratio(a,b) = \frac{|a \cap b|}{\left(\frac{|a| + |b|}{2}\right)}$$

**Figure 1. Equation used for calculating proximity ratio for two mobile phones where *a* and *b* are the sets of cell towers seen by each phone.**

## 3.2 Evaluating Cell Tower Ratio Algorithm for Proximity Detection

Evaluating our *proximity_ratio* algorithm was less than straightforward. It was difficult to obtain a suitably large and appropriate dataset for modeling two stationary phones at a variety of locations. Ideally, we would have simultaneously recorded readings from many stationary phones all at different locations with some ground truth measurement. However this would be hard if not impossible to achieve for a large number of phones. Instead, we used the dataset collected by Chen et al. from their wardriving of Seattle [11]. During this process, they collected cell tower data along with GPS coordinates by driving around the greater Seattle area, equipped with a laptop, 2 mobile phones per carrier and a GPS device. They sampled the phones and GPS device approximately once per second to record cell towers seen by the phones and GPS coordinates. Since two phones were used per carrier, valid comparisons could be made between cell tower readings seen by the two different phones. We used readings from a *Downtown* area with an average cell tower density of 66 towers/km$^2$ and a *Suburban* area with an average cell tower density of 26 towers/km$^2$.

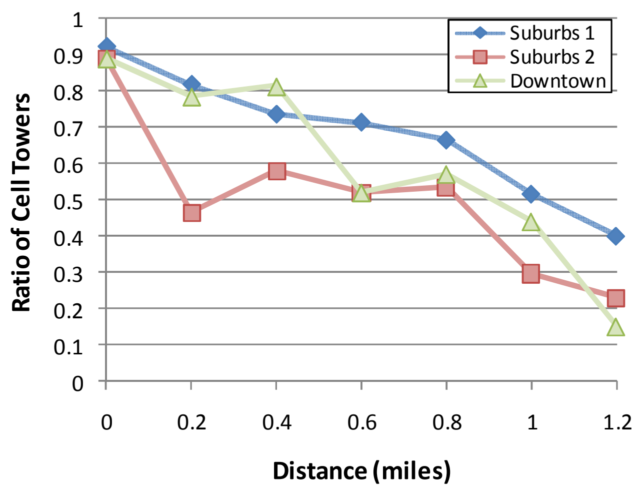


**Figure 2. Initial cell ratio measurements taken from 3 different areas of varying population density.**

This dataset is not entirely applicable to buddy proximity detection. Because the data was collected from a moving vehicle, it only allows for modeling proximity situations where both mobile phones are moving quickly. We are most interested in scenarios when both phones are stationary or nearly so. Still, the data informs scenarios in which users may be moving. When people are driving, they are less likely to be interested in nearby buddies that are also moving, since it is unlikely that both will be available and have free time. In this case, lower recall rates are desirable. If two users are in actuality near each other, this is likely to be temporary and thus a system should not detect it.

Moreover, not all the points in the data set were collected at the same time, with some readings collected almost 5 hrs apart. Due to the load balancing employed by cell towers, comparing proximity between two phones seen at two different times is not an accurate model for a real-time application. To address this, we crosscut the dataset in different ways to approximate the precision and recall of the *proximity_ratio* algorithm for different scenarios. We then consider tradeoffs between precision and recall for different cutoff ratios in these scenarios. We were particularly interested in the behavior for two scenarios: when phones were at the same location, and when they were near each other. By breaking the analysis into these two different scenarios, we can use this dataset to evaluate our algorithm for a variety of distances.

### *3.2.1 Same Location*

To analyze behavior when phones are in the same location (within 100m) and when participants are lingering in areas near each other, we extracted pairs of cell tower data where the readings were taken within 5s of each other. This yielded *28,625* pairs from *Suburb* and *19,087* pairs from *Downtown*. Analysis of GPS readings confirmed 99.9% of these points were within 100m of each other. We then calculated precision and recall numbers based on calculations of ratios for these comparisons. Table 1 shows recall values for different ratios in the *Downtown* and *Suburb* areas. Recall is higher for low ratios and tapers off for ratios between 0.3 and 0.4. Precision is 99.9% since the subset falls within 0.1km.

**Table 1. Recall for different ratios with a distance threshold of 0.1km when phones are at the same location. Precision is 99.9% since the subset falls within 0.1km.**

| Ratio | Recall (*Downtown*) | Recall (*Suburb*) |
|---|---|---|
| 0.1 | 0.96 | 0.96 |
| 0.2 | 0.84 | 0.85 |
| 0.3 | 0.83 | 0.83 |
| 0.4 | 0.57 | 0.58 |
| 0.5 | 0.44 | 0.44 |

### *3.2.2 Evaluating Near Each Other*

We were also interested in situations where two mobile phones were near each other but not necessarily right next to each other. Since the data points collected from this set were taken from the same car, the phones were always next to each other at any particular time. Thus there was no way of getting same-time data from two phones that were far apart. To approximate situations where phones are near one another, we extracted pairs of readings taken within 90s of each other resulting in *569,264* pairs from the *Suburban* dataset and *379,285* pairs from the *Downtown* dataset. We then calculated the *proximity_ratio* for these pairs. Despite the higher recall rates for low ratios reported in the previous section, we knew from our initial studies that low ratios would have a much lower recall in a realistic setting since they detect phones that are miles away as “near” as well, so we focused on ratios higher than 0.3, since this is when recall rates were seen to decrease in the analysis of phones in the same location.

Figure 3 and Figure 4 show the precision of the *proximity_ratio* algorithm when different threshold ratios are used for near/far determination. For phones in this *near each other* scenario, the lower the ratio, the lower the precision. The precision for *Downtown* is higher than that for *Suburb* which is not surprising considering the higher cell tower density in this region. The range of

**Table 2. Recall rates at distances 0.1km to 1.0km for different ratios when phones are nearby.**

| Ratio | Recall (*Downtown*) | | Recall (*Suburb*) | |
|---|---|---|---|---|
| | 0.1km | 1.0 km | 0.1 km | 1.0 km |
| 0.3 | 0.74 | 0.66 | 0.67 | 0.66 |
| 0.4 | 0.50 | 0.41 | 0.42 | 0.40 |
| 0.5 | 0.39 | 0.30 | 0.30 | 0.29 |

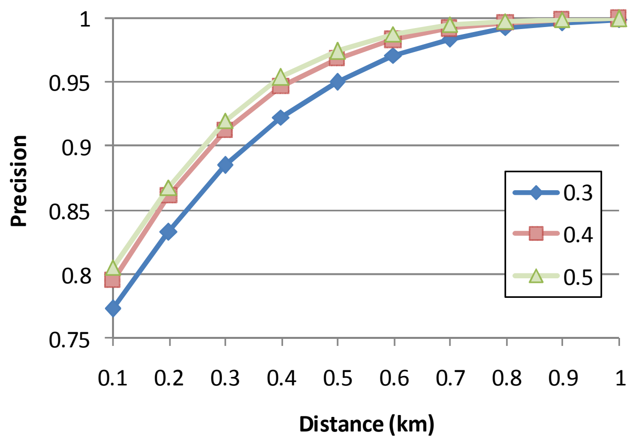


**Figure 3. Precision for different "nearby" distances in *Suburb*.**

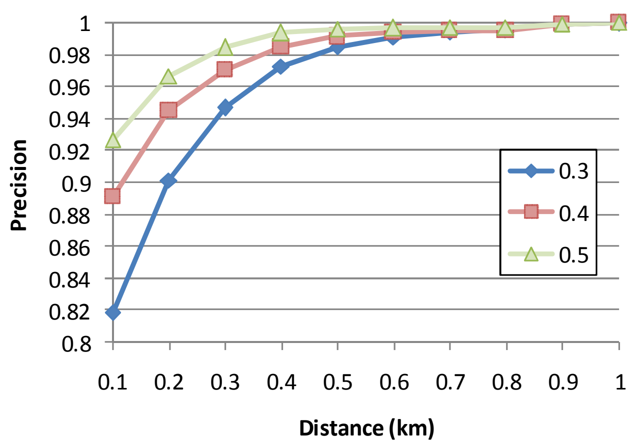


**Figure 4. Precision for different "nearby" distances in *Downtown*.**

recall rates observed over this area are shown in Table 2. While lower ratios still have higher recall rates than higher ratios, this comes at a cost of precision. Considering the design requirements for a peripheral cue application discussed earlier, precision is more important than recall. The issue of how close is close enough is addressed later in section 9. Precision increases as the distance threshold is increased since more false positive results become true positives. It should be noted that precision hits 99% at 0.5km in the *Downtown* case and at 0.7km in the *Suburb* case suggesting that even when false positives are delivered, these false positives are within 0.5km and 0.7km for *Downtown* and *Suburb* locations.

### *3.2.3 Far Apart*

To validate *proximity_ratio* for phones that are far from each other, we decided to look at the entire dataset, even though we knew the temporal problems we described earlier would confound our analysis. For scenarios when phones are far apart, we were particularly interested in low recall while maintaining precision. Specifically, we wanted to make sure that increasing distance would not result in more false positives.

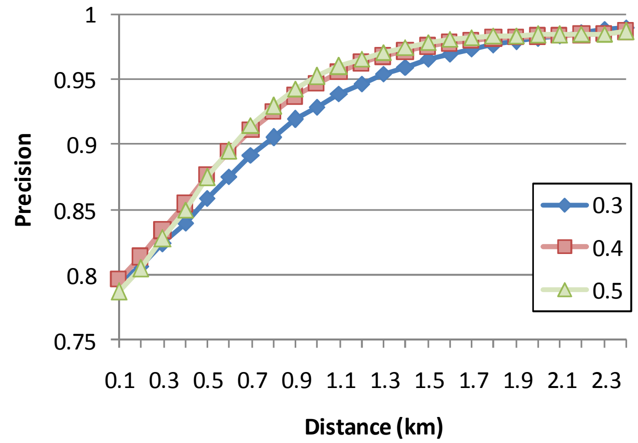


**Figure 5. Precision at different distances in *Suburb* for the entire data set.**

By comparing all of the pairings of readings from one phone to readings from the other phone, we obtained *55,181,015 Suburb* pairs and *36,769,390 Downtown* pairs. Figure 5 and Figure 6 show the precision at different distances for this set of comparisons. The precision values for this data follow the same trends observed before. Precision is higher for higher ratios and all ratios are able to obtain precision of 99% by some threshold (1.0km in *Downtown*, 2.4km in *Suburb*). These findings confirm that the *proximity_ratio* algorithm is effective at reducing false positives for moving phones that are far apart. However, recall rates are significantly lower (Table 3). The rates from the previous sections are much higher and based on more relevant data.

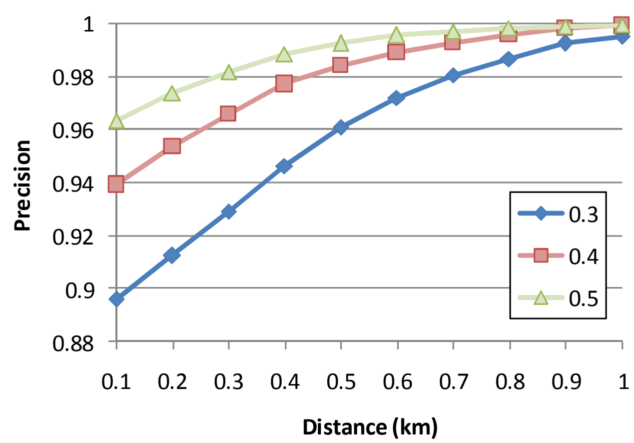


**Figure 6. Precision at different distances in *Downtown* for the entire data set.**

In our experience, the actual precision is lower than that calculated with this dataset and the recall is higher. For the reasons described earlier, it is difficult to gather the cell reading data from many phones simultaneously that we would need for a valid model of phones that are stationary at different distances. We defer qualitative analysis of *proximity_ratio* to a user study described later.

**Table 3. Recall rates for different ratios in the entire dataset.**

| Ratio | Recall (Downtown) | | Recall (Suburb) | |
|---|---|---|---|---|
| | 0.1km | 1.0km | 0.1km | 2.4km |
| 0.3 | 0.12 | 0.08 | 0.09 | 0.06 |
| 0.4 | 0.05 | 0.04 | 0.03 | 0.02 |
| 0.5 | 0.03 | 0.02 | 0.02 | 0.01 |

## 3.3 Sensor Noise

GSM readings can vary widely from moment to moment in ways unrelated to the phone's proximity to the cell towers in the region. This creates the possibility of false proximity detection. Additionally, if buddies hover around a 0.2mi distance from each other for a prolonged period of time, multiple cues might be triggered, creating an annoyance. To mitigate such errors, we implemented a client-side filter for removing sensor noise.

We utilized a proximity reporting mechanism whereby a friend's nearby state is updated only after a number of consistent, consecutive readings. We originally considered using a straightforward approach of waiting until we detected 2 consistent consecutive readings (*2-same-filter*) or 3 consistent consecutive readings (*3-same-filter*) of near or far before updating a buddy's nearby state. In a pilot study, we found that *2-same-filter* helped reduce sensor noise, but still produced a number of false positives. In many situations, when buddies are near a distance corresponding to the ratio threshold of proximity, the ratio readings fluctuate between near and far quite a bit. As an improvement we decided to use a state machine approach, motivated by the 2-bit counters used with branch predictors in computer architecture [33]. Figure 7 illustrates the logic used for this approach. Buddies are initially reported as far away. Edge transitions represent a sensor sampling, yielding near or far. The state of a buddy is only updated to far or near when the states "Report Far" and "Report Near" are reached. This approach could potentially be applied to any binary decision-making sensor as long as it is accurate more than 50% of the time. Henceforth we call the 2-bit counter approach *2-bit-filter*.

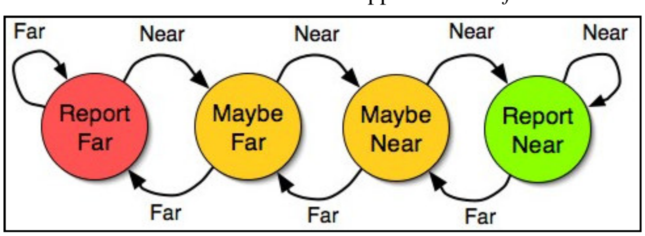


**Figure 7. Two-bit counter for eliminating noise in proximity detection.**

*2-bit-filter* attempts to improve upon these simpler algorithms. 3-same-filter further reduces noise over *2-same-filter* but at the expense of added delay. In the worst case, *2-bit-filter* behaves like *3-same-filter*. However, *2-bit-filter* improves upon this approach because in all but the worst case, it has the responsiveness of a *2-same-filter* with the consistency of *3-same-filter*. As a result it is more robust than either of these two techniques.

To evaluate *2-bit-filter*, we compared its performance against *2-same-filter* and *3-same-filter*. We also used a *baseline* condition whereby we would report *near* or *far* based on a single reading. For sensor noise filtering, we were interested in situations where users would be transitioning from *far* to *near* or vice versa. To extract these scenarios from the original dataset, we extracted readings from the original dataset at 30s intervals. We then ran the three algorithms and the baseline on the resulting dataset.

Since all ratios showed a similar reduction in false positives, we report on the average reduction over the baseline. The usage of a *2-same-filter* was effective at reducing noise, reducing the average number of false positives by 53.8%. *3-same-filter* reduced false positives by 80.9%. 2-*bit-filter* was most effective, reducing false positives by 84.9%. The reduced false positives translated into a higher precision, increasing by an average of 5% for a distance of 0.1km. This is quite significant considering the already high percentage of precision being reported in the previous section. Recall rates were improved by a negligible amount.

This analysis demonstrates that the *2-bit-filter* can be an effective technique for reducing sensor noise by maintaining recall while reducing the number of false positives. Not apparent in this analysis is the cost of using one of these filtering schemes, namely delayed proximity detection, which we address in the next section.

## 3.4 Minimizing Power Consumption

The limited power supply from mobile phone batteries requires careful consideration in a continuously running context-aware system. We addressed this by adjusting the sample rate and by minimizing unnecessary transmissions over the data network.

Sampling cell towers quickly often did not yield a change in seen cell towers, suggesting we could reduce the sample rate to save power, without dramatically affecting proximity detection response time. To get an idea of how sampling rate would affect power consumption, we originally chose a sample rate of 1 sample/20s. This caused the phone battery to discharge in less than a day, unfeasible for a study in the wild. To address this we decreased the sample rate to 1 sample/90s which turned out to be sufficient for a study in the wild, only requiring a recharge every other day. However, the usage of the 2-bit counter described in the previous section introduced a potential delay of 3 sample periods, 270s at a sample rate of 1 sample/90s.

To reduce the delay of proximity detection, an adaptive sampling rate was used. Initially, buddy proximity is sampled at a rate of 1 sample/90s. When the counter moves into a "maybe" state, the sampling rate is increased to 1 sample/20s, until steady state is reached (either "Report Far" or "Report Near"), at which point the sampling interval reverts to 1 sample/20s. This approach mitigates the delay of proximity detection reducing it to approximately 1.4 times the original sample rate. Initial data collection suggests *2-bit-filter* used in conjunction with an adaptive sampling rate provides good filtering of noisy data while reducing the delay of proximity detection. To avoid redundant notifications for buddies hovering around the near/far cutoff, the cues for a pair of buddies are delivered at least an hour and a half apart.

Our use of two sampling rates helps reduce power consumption to some degree, but measures are needed for situations with poor network signal. For one, sending data over a poor link tends to consume more battery power, in part because these data transmissions are more likely to fail, causing the underlying system to continue to attempt sending. Two, a poor link is indicative that there are no cell towers that are strongly suggestive of the phone's relative location, so making a report provides no information about the phone's whereabouts. These black hole situations are common in the USA, such as inside buildings with lots of metal or concrete. The PeopleTones client detects these situations by comparing the phone's signal strength to a threshold. To compensate for when clients in these situations do not update, the server retains the last reported reading from the phone along with a timestamp, so that others can still make inferences about proximity for a while, assuming that the non-reporting of their buddies is caused by being in a building.

To measure power consumption, we timed how long it took to drain a fully charged iMate SP3i (HTC Tornado) running PeopleTones with the considerations described above. It took 2 days and 16 minutes to fully discharge the battery. This time period was deemed sufficient, precluding the need for daily recharges.

# 4. PERIPHERAL CUES IN THE WILD

For the purposes of this study, we made three considerations to keep cues unobtrusive:

- First and foremost, the cue should not invade the periphery.
- Second, when the cue is perceived, it should not be seen as inappropriate in any way, most notably by those for whom the cue is not intended – a matter of etiquette.
- Third, because the periphery is constantly shifting with one's attention, perhaps as demanded by other changes in the environment (e.g., someone speaks to you, or shifting traffic conditions while driving), the cues, when perceived, should not be distracting – they should not impede shifts in attention or other natural changes to the periphery. In particular, people should not have to think about the cues that they are perceiving.

We refer to these three properties collectively as *unobtrusiveness*. With these issues in mind, the principal challenge with the use of peripheral cues in the mobile setting is resolving the tension between reliable receipt of cues and unobtrusiveness, without making unrealistic assumptions such as the required use of headsets. Peripheral cues can be overlooked without harm, and as designers we can err on the side of cues being missed.

With these considerations in mind, we decided to use a ratio threshold of 0.4 for our peripheral cue application. Based on our findings reported earlier, this provided good precision while maintaining fair recall. Our pilot studies confirmed this was an effective ratio for the deployment area.

We hoped to gain insight on these complex considerations over the course of our study, but we did have some initial hypotheses. One, short audio cues would be less invasive and more polite than long cues. Two, having corresponding vibration cues could be useful both for politeness and increasing chances of being perceived in noisy environments. Three, environment sensing could support the adaptation of the cues being played to ensure consistent maintenance of peripherality and politeness.

## 4.1 Auditory Cues

Since much past work with peripheral cue systems has used sound cues to deliver information, we followed in suit. Playing sound cues from a mobile phone is natural, but has potentially different requirements than environmental-based systems. Past work has found that short, rich auditory cues that build off of sounds users are accustomed to hearing in their normal lives can provide information to users serendipitously [32]. We explored a number of different types of sound cues. Soothing nature ecologies have often been used and so we created a set of nature cues of 3-5 seconds in duration. Music cues were also explored given that music timbres are effective for conveying information [6]. Many mobile phones have the ability to map specific ringtones or music clips to different users on a contact list. While many people use these, the efficacy of mapping sound clips to identity is relatively unexplored. Yet, for a buddy proximity application, music clips seem promising for mapping the identity of a person to an audio cue, given the possibility of a semantic link [13].

## 4.2 Vibrotactile Cues

In many office setting studies of peripheral cues, a headset or other wearable device is often the delivery mechanism used for delivering auditory cues in an etiquette-friendly manner. When delivering peripheral cues in the wild, where the user can be in a variety of social settings, it is unreasonable to require them to wear an additional device for receiving auditory cues. Mobile phones offer the ability to play sounds using their speakers, which can be effective for informal situations, but it is unlikely that this delivery channel will always be socially acceptable. Much like the silent or vibrate-only modes on mobile phones, peripheral cues delivered via these devices must also have a socially etiquette friendly mode [17].

Motivated by haptics research suggestions to use vibrotactile cues for ambient information delivery [35], we explored using vibrotactile patterns to convey ambient information on mobile phones with the actuator that commonly ships with these devices. Ideally, there would be a one-to-one mapping of sound cues to vibrotactile patterns, where a user could easily identify a vibrotactile cue and its respective auditory cue. However, generating a variety of distinguishable vibrotactile cues can be difficult on commodity mobile phones, given the limited API; most mobile phones only support the functionality of turning the actuator on or off. With the exception of phones with specialized built-in hardware [23], the API for most phones does not support playing vibrotactile pulses of different amplitudes nor do they provide any low-level functionality to specify the amount of current used to drive these actuators.

### *4.2.1 Generating Different Vibration Levels Using Mobile Phone Actuators*

We present an algorithm to generate a wider range of vibrotactile sequences that circumvents API constraints on actuator functionality. While a full analysis of the capabilities of this approach is outside the scope of this paper, the basic algorithm for playing a pulse of varying amplitude is presented below for completeness.

By changing the duty cycle[1] of the voltage sent to the motor, different speeds can be obtained. Similar techniques are used to reduce power consumption of DC motor. This approach also reduces motor speed, making it useful for our goal of modulating the level of vibration. Our software approach repeatedly turns the actuator on for short periods of time, spinning between calls to the function that turns the actuator on. Timing is critical during this process, so the active thread is given the highest priority to avoid inopportune context-switches. By doing so, we demonstrate that we can achieve pulse-width modulation[2] via software. Different amplitudes can be generated by varying duty cycle.

```
vibeLength = 20;
onTime = 1;
offTime = 9;
endTickCount = currentTickCount() + vibeLength;
while(currentTickCount() < endTickCount)
{
    playVibrate(onTime);
    sleep(offTime);
}
```

**Figure 8. Code for generating a 20ms vibrotactile pulse.**

Figure 8 shows a code segment for this process. The `playVibrate` function represents the standard function for turning the actuator on, supplied by almost all mobile phone APIs. The variables `onTime` and `offTime` control the amount of time the actuator is turned on and off respectively. We demonstrate that by changing the values of these, we can adjust the duty cycle of the vibrotactile actuator, changing the level of vibration generated.

A series of pilot studies found that people could not detect pulses played for less than 20ms in this manner and suggested the operating range could be divided into 10 differentiable *levels*, sufficient for this study. To generate a 20ms pulse level of 1, values `onTime=1`, `offTime=9` are used. To generate a 20ms pulse level of 9, `onTime=1`, `offTime=1` values are used. A pulse level of 10 is generated by calling the `playVibrate` function for the desired pulse length. Using this approach, a vibrotactile pattern can be defined as a sequence of such pulses of varying level.

To examine the effects of our software approach on the actual hardware, we opened a phone and measured the voltage levels produced by our software using an oscilloscope. Figure 9 shows sample oscilloscope plots measured for 4 different amplitudes. These plots confirm that our algorithm successfully achieves pulse-width modulation and that the different levels of vibration produced are the result of this. Mapping Sounds to Vibrotactile Patterns

With peripheral cues deployed in the wild, a number of situations will arise where auditory cues will be socially disruptive (e.g. during a meeting) or might not be heard over ambient noise (e.g. walking by a busy street). We generated vibrotactile cues as a complement to auditory ones, hoping to leverage the association of auditory cue and buddy identity. If a vibrotactile pattern can be generated such that users can match it to its corresponding audio cue, then users can map the vibrotactile pattern to the buddy cue as well. This would reduce the need for learning a vibrotactile language.

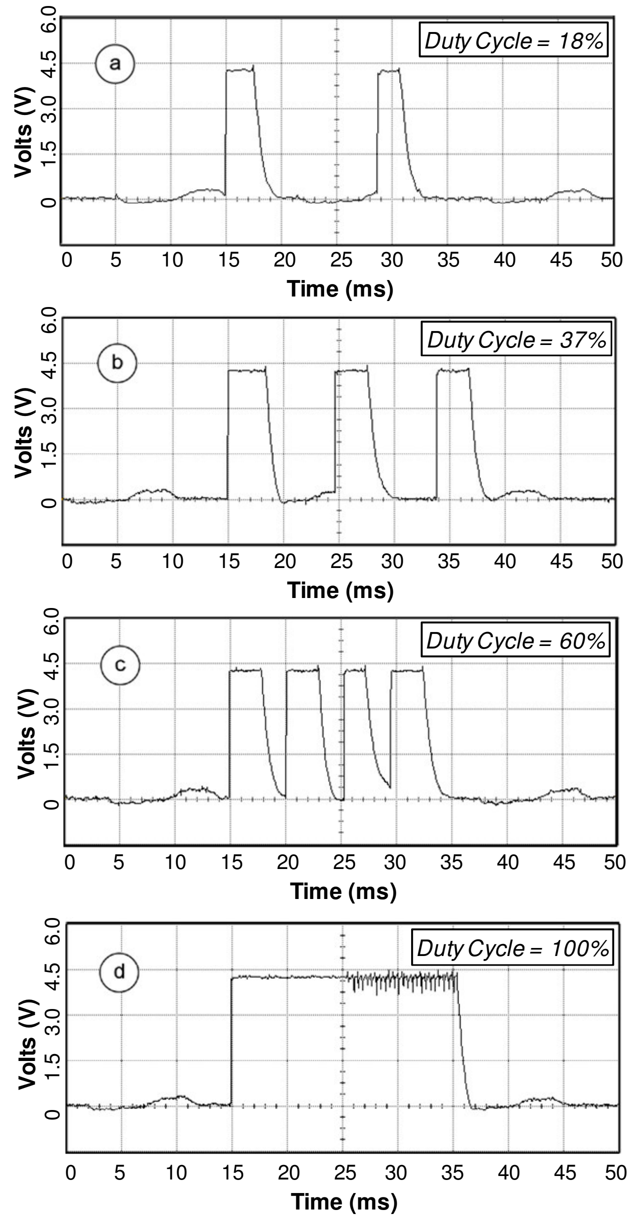


**Figure 9. Oscilloscope plots of voltage generated for vibrotactile actuator for pulses at levels (a) 1 (b) 3 (c) 5 and (d) 10.**

Mapping auditory cues to vibrotactile sequences is challenging. On the one hand, there are difficulties associated with trying to map from an auditory system to a tactile one, where different receptors are being used to receive information [2]. This issue becomes further complicated by the significant differences in sample rates. Our pilot studies found that participants had difficulty differentiating between vibrations separated by less than 20ms. This generates a signal with fidelity equivalent to a signal sampled at 50Hz. A typical music file is sampled at 44.1kHz, a full three orders of magnitude greater, capable of capturing far more fidelity. To address this gross level of under sampling, we utilized a

[1] Duty cycle refers to the proportion of time that the device is turned on.

[2] Pulse-width modulation refers to the modulation of duty cycle.

number of digital signal processing techniques as part of the encoding process to try to capture the essence of the sound. We used a semi-automated method for converting a song to its vibrotactile equivalent using Matlab on a desktop PC.

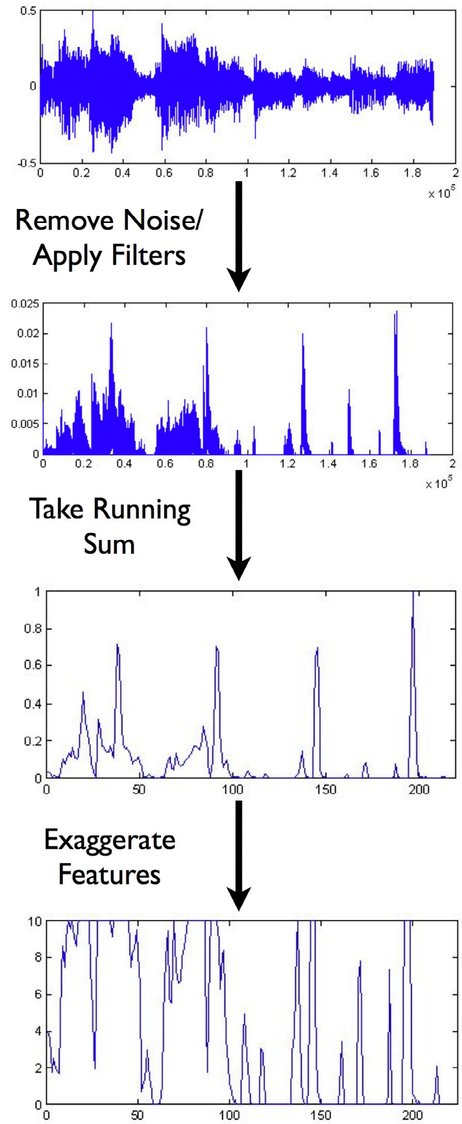


**Figure 10. Block diagram showing the process of converting a wave file to a vibrotactile pattern.**

Capturing the essence of a song is a known hard problem [1]. Initially, we considered the beat of the sound by examining lower frequency components of the clip. While this can be effective for certain sound clips, our experience suggests that it is because of the higher amplitudes of the low frequency components in those sequences. The lyrics of the song chorus were also thought to be important to characterize, given their use in identifying songs. However, in practice, lyrics are difficult to map to our vibrotactile language due its lower fidelity.

Pilot studies suggested that a combination of amplitude thresholding and bandpass filters would be the most promising approach. While lyrics are important in recognizing songs aurally, they are in practice difficult to map to vibrations. Instead, we aimed at mapping the beat of the song to vibrotactile patterns. We also found that by exaggerating the difference between loud and quiet sounds, the song was better characterized. The general process can be thought of as trying to create a humming sequence for the audio clip. Figure 10 outlines the general steps of this process.

The first step in converting a sound file into a vibrotactile pattern is to remove noise from the original signal. In this context, we consider "noise" to be elements of the sound that are not significant to the vibrotactile encoding of the sound, in addition to the traditional definition of the term. Our pilot studies found that components of the signal falling between the frequencies 6.6kHz to 17.6kHz were a good balance between noise reduction and keeping the original signal. We used an $8^{th}$ order implementation of the Butterworth Filter (a commonly used filter for band-pass filtering [18]) to isolate the components of the signal in this frequency range (Figure 10-Remove Noise/Apply Filters). Additionally, we use an amplitude threshold to remove components from the output of the bandpass-filtering step. We only keep components that are greater than the average of the output.

The next step in the process is to try to characterize the resulting processed signal in a way that preserves the characteristics of the sound file. To do so, we take a running sum of the absolute values from the output from the previous step, generating 1 value for every 20ms (Figure 10-Take Running Sum). Each sample now represents a value that can be played for 20ms while keeping length of vibration and length of sound clip consistent. Finally, the differences between loud and quiet components of the signal need to be exaggerated (Figure 10-Exaggerate Features). We do this by composing the output from the previous step with a power function of the form $Ax^n$ where x is the sample value and $A$ and $n$ are constants in the ranges: $10 \leq A \geq 15$, $1 \leq n \leq 2$. Part of the reason this is currently semi-automated is because we used different constants for different songs. Generally speaking, we used larger values of $n$ when there was a larger range of frequencies in the original sound, and smaller values of $A$ when the signal was louder. The result is a sequence of values representing a vibrotactile pattern that preserves many of the characteristics of the original sound signal.

# 5. PEOPLETONES

To validate the system level components we described above and to explore peripheral cues in the wild, we developed PeopleTones, an application for informing users of buddy proximity via peripheral cues from their mobile phones. A sound clip and corresponding vibrotactile pattern is associated with each buddy.

To inform the user of a buddy's proximity, the user can specify to have only vibrotactile cues, only audio cues, or both be played by selecting the appropriate phone profile. Alternatively, the user can enable an automatic noise detection mode to select an appropriate form of delivery. On the one hand, sound cues delivered in the middle of a meeting can be disruptive; on the other hand a sound cue delivered in the middle of a loud concert would be futile. Based on prior research, we knew that detecting interruptability on a mobile phone would be impractical at best [36], but detecting the noise level in an environment using the phone's microphone to adjust the "level" of the cue is practicable. At the minimum, we hoped to learn how people would react to a mechanism for automatically choosing the mode of cue delivery.

When a cue was triggered to be delivered, ambient noise level was measured for 5 seconds and its average amplitude computed. In quiet environments, only the vibrotactile cues were played. In loud environments, both vibrotactile and sound cues were played, as vibrotactile cues can be felt in noisy environments when even loud sound cues might be inaudible. Quiet and loud thresholds were calibrated using both a quiet office environment and the student center of a University during a busy hour.

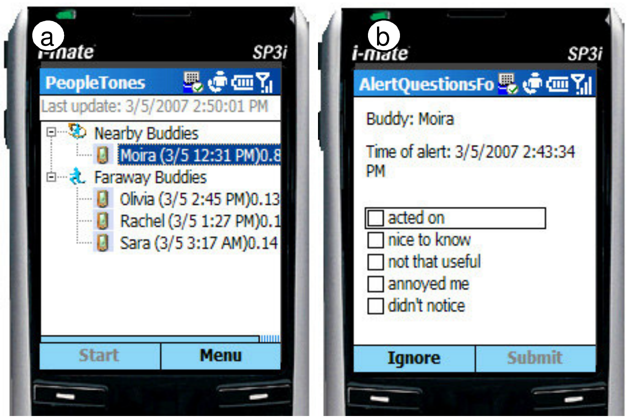


**Figure 11. (a) The main PeopleTones user interface showing buddies that are near and far, as well as the last time the system was updated. (b) Post-alert prompt asking participants how useful the alert was.**

PeopleTones is implemented as a client-server application using standard SOAP web services. The client-side application is written in C#.NET on the Windows Mobile Smartphone platform. The interface is shown in Figure 11a. Each phone periodically pushes its GSM cell tower readings to the server, which computes buddy proximities and then notifies the phones of changes using the techniques described earlier. In situations where a client does not send the server an update of their location, the location uses a timestamp along with a copy of the last update received from that user. While all the users in our study used the same adaptive rate, this architecture allows clients to update their location as often as they like, allowing them to make their own power considerations.

# 6. USER STUDY

We performed a naturalistic study by deploying PeopleTones to three groups of friends, forming three different test conditions, each for the course of two weeks.

**Table 4. Group makeup for the three groups.**

| Group | Group Size (Gender) | Age Range | Makeup | Condition |
|---|---|---|---|---|
| **Nature** | 5 (F) | 19-21 | Roommates | Nature Sounds |
| **Your Choice** | 8 (5F, 3M) | 22-26 | Friends From Church | *You* choose what I hear |
| **My Choice** | 4 (1M, 3F) | 19-22 | Close Friends | *I* choose what I hear |

The purpose of this study was two-fold. First, the results from previous sections suggest that *proximity_ratio* when used in conjunction with *2-bit-filter*, should have high precision and modest recall. Yet, we wanted to test the hypothesis that, due to people lingering at places where they work, live, and play, that our participants would experience both high precision and high recall. Second, we sought to understand how peripheral cues worked in the wild, especially as regards obtrusiveness, comprehensibility, and the behaviors that resulted from their use. Thus, the three conditions were varied by the kind of peripheral cues that were employed.

# 7. PARTICIPANTS

We recruited three groups of friends forming groups of sizes 4, 5 and 8 people. These 17 participants consisted of students and young working professionals, 12 women and 5 men, aged 19-26. Participants were recruited based on interest in a buddy proximity application as well as having physically proximal friends. Participants were given an American Express Gift Card as a thank you for their time.

## 7.1 Methodology

Participants used PeopleTones over the course of 2 weeks. We conducted 4 interviews over the course of the study. Prior to the study, a pre-study interview was conducted to gather basic demographic information, mobile phone usage habits and general "closeness" to the participant's friends whom were also participating in the study. Additionally, a pre-study was conducted to evaluate whether participants could match the semi-automatically generated vibrotactile patterns to sound clips. A mid-study evaluation was also conducted to make sure there were no problems with the system. Finally, a post-study interview was conducted to reflect on the participant's experience. A test of matching vibrotactile patterns to music cues was again performed to measure learning effects, if any, that may have taken place over the course of the study, and to evaluate consistency.

The three groups of friends formed three different conditions for cue-to-information mapping methods. Group Nature (N) consisted of 5 friends who were given a set of nature sounds to assign to their friends. However, they opted for automatic assignment of cues, since they felt there was no relationship between the cue and their friends. This eliminated the need for a full 2x2 study whereby a group of users who selected their own Nature cues would have been included. Group Your Choice (YC) consisted of 8 friends who selected a single sound for themselves, representing the cue that their friends would hear when they were nearby. Group My Choice (MC) consisted of 4 friends who each selected the cues that they would hear, when their friends were nearby. Table 4 summarizes these group conditions.

Before the study, participants identified music cues that they would want to use for the study. They were given the option to identify specific parts of the song that they wanted to use. Alternatively participants could select from 2-3 different 3-5 second segments of the song selected by the authors, typically chosen for their mapability to vibrotactile patterns. After participants selected song segments they wanted to use, a corresponding vibrotactile pattern was generated, using the procedure described in Section 4.

# 8. USAGE AND SELF-REPORTED DATA

To perform data analysis, we performed client-side logging when a cue was triggered. Once a cue was triggered, the participant was presented with a form asking them if they acted on it, if it was nice to know, if it was not useful, or if it was annoying (Figure 11b). Alternatively, they had the option to ignore the form if they were busy. This post-cue questionnaire was left on the screen so they could later respond. If they did not choose to ignore the cue, they were presented with a form asking them if they could tell who it was to which they could respond "yes from sound" "yes from vibration "yes,other" or no. In this case, "yes, other" was intended for situations where the participant knew who it was based on other factors (e.g. knew their roommate was coming home around that time) This was recorded for completeness and not factored into the comprehension rates described below. Forms were left on the screen until they were responded to.

A total of 683 cues were sent over the course of 2 weeks, across all conditions with 122 cues in the Nature group, 466 cues in the Your Choice group, and 95 cues in the My Choice group. Each cue resulted in a post-alert form being displayed. Using self-reported forms displayed on the mobile phone, the user was queried both for their response to the cue as well as whether they could identify buddy that the cue represented. The breakdown for these post-cue responses is shown in Table 5 and Table 6 respectively. Since all cues elicited a response form and all forms required a response (even if the response was "Ignore"), the percentages are also reflective of the 683 total.

| | Acted On | Nice To Know | Didn't Notice | Not That Useful | Ignored | Annoyed |
|---|---|---|---|---|---|---|
| Nature | 4% | 38% | 22% | 25% | 10% | 1% |
| Your Choice | 9% | 34% | 31% | 5% | 20% | 1% |
| My Choice | 12% | 60% | 15% | 2% | 8% | 3% |

**Table 5: Self-reported response to the cue.**

| | Yes From Sound | Yes From Vibration | Yes Other | Ignored | No |
|---|---|---|---|---|---|
| Nature | 3% | 2% | 17% | 2% | 76% |
| Your Choice | 76% | 0% | 7% | 1% | 16% |
| My Choice | 64% | 16% | 3% | 3% | 14% |

**Table 6: Self-reported identification of the cue's information.**

# 9. DISCUSSION

In the following section we reflect on our two major research questions: the suitability of peripheral cues as an in-the-wild communication mechanism and the suitability of mobile phones for providing such cues. We draw on the observations and data above, as well as from our interviews with the study participants.

## 9.1 Peripheral Cues are a Viable Communication Mechanism in the Wild

Unobtrusive peripheral cues in the wild, while challenging, can be achieved by informed cue design and providing personal control over cueing mechanisms.

### *9.1.1 Designing and Choosing Cues for the Wild: Music and Personal Control*

Although office-setting studies have found soothing nature ecologies to be effective for comprehension and unobtrusiveness, cues in the wild should be composed of music, and perhaps repeated.

With regards to comprehension, the self-reported usage data shows that groups Your Choice (YC) and My Choice (MC) both demonstrated an 83% comprehension rate, where comprehension is defined to be when the user could identify the buddy from the cue, collapsing results from audio and vibration (Table 6). In contrast, the nature group demonstrated a significantly lower rate of 22%. Interestingly, this lower rate did not result in lower usefulness ratings (42%) for the application when compared to the Your Choice group (43%). Perhaps the ability to look at the phone after receiving a cue mitigated the negative effects of cue comprehension. Many participants cited that they would prefer longer cues since they could be difficult to catch in the dynamic environments of their daily lives. For example, <MC-1> commented: *"sometimes couldn't hear because the song was too short."*

The obtrusiveness of music cues was not a concern. The reasons are somewhat surprising. <MC-3> comments: "*When it went off in [the library] it didn't actually seem to annoy other people too much, they just thought it was just another phone*." This observation points to the fact that mobile phones have become largely invisible and socially accepted, at least for young adults, even in a "quiet zone" like a library. (We reflect more on etiquette concerns in the next section.) Another reason cited for the unobtrusiveness of music cues was the positive feelings generated by the music. <MC-1> comments: "*I would like longer songs so I could hear it and because I like the songs*." The Your Choice group made similar comments, even though they did not pick their music cues. <YC-5> comments that she liked: "*Just hearing the songs. I liked the fact that each person could choose whatever they want for their own identity. Since it was a small group of us, it's kind of fun and it felt like this is a group of us*." Overall, 9 of the 12 music participants volunteered a liking for hearing music. Interestingly, it appears that cues with emotionally positive associations are generally unobtrusive.

Music cues are similar to the ringtones sometimes used for caller ID on mobile phones. However, ringtones in the wild are relatively unexplored. <YC-3> comments: "*It was fun how everyone had a song specific to them. Adds a little bit of personality. I don't use ringtones, that's why it was neat for me. Too much trouble to do on my phone*." The results of this study validate the usefulness of ringtones, in being able to successfully convey information about people in a pleasurable way.

The usage of music cues also seems to reinforce learnability, with 83% of users in conditions My Choice and Your Choice being able to identify who the cue was for, based on self-reported post-notification questions. This learning effect is also reflected to some extent with vibrotactile patterns. The My Choice group was the only condition with an appreciable amount of cue identification from vibration, demonstrating 25% identification rate from vibration cues alone. While not overwhelming, this acts as a proof-of-concept for the delivery of ambient information via low fidelity haptic channels. Analysis of the before and after vibration studies suggest some users are consistent in the way they match vibrotactile patterns to sound, with 7 participants responding consistently, when comparing their before and after responses. 75% of My Choice was able to correctly map vibrations to sounds and then to people. Participants in the Your Choice condition were less successful in mapping vibrotactile patterns to music, possibly because of the larger number of cues or because they were not as familiar with the songs selected for cues. When presented with the task of matching vibrations to sounds in the post-study interview, participant <MC-3> exclaimed "*Oh that's Cathy!*" when she felt the vibration associated with the music cue associated with Cathy. When comparing error rates for a matching vibration-to-sound task from before and after studies, minimal improvements were observed, suggesting minimal learning effects.

This does not necessarily suggest that music cues should always be used over nature cues. Rather, it reinforces the idea that when designing peripheral cues, it is important to maintain a semantic link between the cue and the content being delivered. Nature cues could be useful if they had some meaning when associated with the object of interest. In the case of buddy peripheral cues, participants were much more likely to have semantic associations between music and friends.

### *9.1.2 Personal Control over Cueing Mechanism for Unobtrusiveness*

The discussion above suggests that personal selection of cues aids both comprehension and unobtrusiveness. In addition, for many users, explicit control of the notification modes was important. Although personal control has been cited as important in the design of a number of social mobile systems, these concerns typically have to do with privacy [21]. In our study, personal control over the cueing mechanism was a critical element for controlling unobtrusiveness and interruptibility.

Even though an automatic ambient noise level feature was provided, many users opted not to use this mode, not even trying it before dismissing it. In fact, 12 of the 17 participants did not even try the Automatic mode, despite the fact that the phone was put in Automatic mode when given to the participants. When asked about the automatic mode in the post-study questionnaire, <MC-4> commented "*I didn't use it. I was afraid to use it since my professors this quarter are pretty anal. I kept it mostly on vibrate when I was in class, or in normal mode when I wasn't*." For this participant, personal control of the notification mode was important because they feared PeopleTone cues being triggered in the audio mode in a classroom setting where it might be disruptive. Participant <YC-1> expressed a similar concern, saying "*I was afraid that if I was at church, it wouldn't work. It would just backfire on me and I wanted to have been more sure about it.*" Like <MC-4>, <YC-1> was afraid of unwanted notifications while at church, another social context where an audio notification would be unacceptable. It should be noted that both <MC-4> and <YC-1> considered social contexts where they expected the notifications to be triggered, in this case as defined by their group's shared interests. In addition to a lack of trust in the application's accuracy for detecting ambient noise in high stakes situations, a number of users also expressed uncertainty as to what the system considered to be loud or quiet environments. <YC-4> said "*I'm not too sure what happened or how loud the environment needed to be. I'd want to determine how reliable the function is before using it and to check how often it looks at the environment*." <YC-4>'s comment suggests a potential solution to this problem is for some type of system feedback whereby through manually user-controlled notification management, the application gains the user's trust, demonstrating that it works well in a variety of environments. This could be done with some type of visual indicator in the application of the type of cue that would be delivered, which the user could check while in different environments. Of course, this solution requires the user's visual attention during the familiarization period.

Still, even with a trustworthy system in place, some users had different requirements than those retained by the system. <MC-2> commented "*Sometimes when it's quiet, I don't need it to be quiet, like when I'm at home by myself. I think I felt it one time but I generally like to keep it on [loud]*." The user trusts the Automatic mode to work as expected, she prefers the mode that plays both vibrotactile and audio cues, simply because in certain quiet situations, auditory cues will not be irritating to anyone.

Finally, for some participants, different notification modes offered different fidelities of information. <YC-8> commented that "*Did use the vibrations, but didn't work out well. I felt it vibrate, but I could indicate [who it was] better with sound. The sound lets me instantly figure out who it is. With the vibrate, you have to wait 5-10 seconds to figure out who it is*." Although this comment touches on the issue of learning mappings between the different notification modes, <YC-8> expresses a distinct preference for sound cues, crediting their higher fidelity. Similarly, <Nature-1> said "*I wish it had given me louder alerts*," suggesting that some mechanism for controlling the volume would have been useful for some.

### *9.1.3 User Information Needs: Peripheral Cues Provide an Overview*

When our participants were asked about how physically near friends needed to be to be considered "nearby," many people cited the mode of transportation as being relevant. For people traveling by foot, distances within about 0.5 miles were cited as being "nearby." For people traveling by car, a distance of 2-5 miles was considered nearby. Our results from Figure 5 show that our algorithm was able to achieve precise detection within these limits. Additionally, 15 of 17 participants reported that PeopleTones' implementation of near was good enough for buddy proximity, verifying the results from the dataset analysis.

The proximity algorithm used by PeopleTones, which detected proximity at around 0.2 miles, was accurate enough for user needs. When asked about the accuracy of the system, most participants commented that it detected "near" for a buddy most of the time when the buddy was known to be near, and "far" when far away. Activity detection [40] could conceivably be used to adjust the cutoffs according to one's speed of movement, although the accuracy reported by users suggests that this may not be necessary.

Even though PeopleTones' proximity algorithm was deemed accurate enough, 4 people spontaneously volunteered that they also wanted to know the actual distances of their buddies, and 3 of those wanted the location. "*I'd like if it could tell me exactly how close they actually are*," said <MC-2>. <YC-5> offered that "*The only thing I didn't like about using this phone was not knowing exactly where that person is*." (PeopleTones's proximity ratio is not suitable for computing exact distances or locations, but ratios significantly above the "near" cutoff could be used to infer "close".) This information need is not surprising, since such information would inform, for example, a decision on whether to call the person to arrange a meeting. This information need not be provided by the cue itself. Schneiderman's Visual-Information Seeking Mantra "Overview first, zoom and filter, then details-on-demand" [37] suggests that the peripheral cue should be treated as the overview, with additional information displayed in the user interface providing details on demand. Also, because the cue is an ephemeral overview, and may not be fully comprehended, the visual interface provides valuable redundancy. For this reason, the PeopleTones user interface used by the participants displayed the near/far state of the buddy and the time of the inference. <MC-4> volunteered "*If I wanted to know if anyone was nearby, I liked how it showed the last time it had checked next to their name*."

At the same time, displaying distance could be perceived as an invasion of privacy. Surprisingly, only 2 of 17 users reported privacy as a concern. This could be because the groups of friends were tightly knit, or because exact distance was not shown. The 2 participants concerned with privacy suggested providing user control of who could see their location.

## 9.2 Mobile Phones Are a Viable Platform for Context-Aware Peripheral Cue Applications

As the above results convey, mobile phones appear to be a viable platform for proactive context awareness when there are asymmetric tradeoffs to be leveraged. In the case of an application like PeopleTones, missing an event of interest, whether due to sensing or notification, is acceptable. There were few reports of false negatives and only one report of a socially problematic cue. There were also few reports of false negatives.

Yet, several other factors unrelated to conservative design contribute to the viability of peripherals cues on phones. Mobile phones and the sounds they make are socially accepted in many settings, aiding unobtrusiveness. As personal devices they enable personal control, which aids the comprehension of cues, creates the positive associations that permit managing the periphery, and ensures their unobtrusiveness when silence is paramount.

Additionally, with the use of our novel DSP techniques, the commodity vibration actuators found on mobile phones are an adequate channel for etiquette-sensitive situations. For music-based vibrations, people were very good at matching the vibration patterns to their songs. One group, the My Choice group, found the vibrations to be useful in the wild, serving as the delivery mechanism about 16% of the time.

Likewise, several factors unrelated to conservative design contribute to the viability of context sensing on mobile phones. Direct phone-to-phone comparison of cell tower readings not only achieves ubiquity but also avoids a possible source of error by not calibrating to a third frame of reference (absolute location). Emphasizing the elimination of false positives is apparently effective because the lingering of buddies eventually leads to successful recall. Timeliness is not a critical feature of buddy proximity, but some users did complain about the occasional slowness of the reporting, suggesting that dwelling in a place eventually leads to a positive report.

As corroborating evidence, the participants told many stories about how PeopleTones affected their behaviors or dispositions. Here are a few typical quotes, at most one per participant:

> *"One time at the library, I wanted to eat with someone and so I went outside to call someone. The phone vibrated. I just called the person to meet up."*

> *"Whenever I drive to school I found out where <YC-7> works because I always get her alert when I'm driving on Miramar. Oh, so she works around here?"*

> *"I thought it was so neat every time it would ring. It made me really happy. Oh! They're right here, or oh! They're right there."*

> *"It was cool to see who was home by the time I got home. I could tell if <YC-1> was home when I passed by University. So if we were going to go eat or something I could ask her. Oh she's home, so let's call her and see if she wants to eat."*

# 10. CONCLUSION

Employing mobile phones for proactive context awareness holds promise due to the ubiquity of mobiles and their infrastructure, yet phones' necessarily inexpensive construction presents challenges like imprecise sensors, clumsy actuators, and limited battery life. For the case of detecting and reporting on "nice to know" situations such as the proximity of a friend, the precision of sensing must be high enough to minimize annoying false notifications, and the notifications cannot be too obtrusive to the user or those in the vicinity. We explored these issues through the PeopleTones buddy proximity application.

We have contributed (1) an algorithm for detecting proximity, (2) techniques for reducing sensor noise and power consumption, and (3) a method for generating unobtrusive peripheral cues.

For detecting proximity we compared the cell towers seen by the mobile phones to estimate proximity. GSM's long range and random characteristics means that a phone will report false positives, so we filter the proximity data using a simple state machine. With these techniques we are able to achieve 99% precision for a ratio threshold of 0.4 and fair recall. The counter also manages power consumption by sampling at a slow rate when the state machine is in the typical *far* or *near* states. Power is further conserved by not reporting when the GSM signals are weak.

We took a peripheral cue approach to providing notifications, using both short audio and corresponding etiquette-friendly vibrotactile cues. To achieve a language of corresponding vibrotactile cues, we introduced an offline digital signal processing technique that captures the essence of audio cues, whose patterns are realized on the phone by generating range of amplitudes using a technique similar to pulse width modulation.

Our controlled studies of proximity detection based on wardriving data revealed high precision, especially with the 2-bit counter, but only modest recall. In real world settings, where people dwell at locations for significant periods, recall appears to be much higher because the algorithm has more chances to detect proximity.

The user study revealed that peripheral cues are an effective, unobtrusive mechanism for notifying people of such inferences. Although haptics have often been suggested as a promising ambient delivery mechanism, sound was the preferred medium, possibly because of its higher fidelity. Our method for encoding sounds into vibration patterns on the limited vibration motors of mobile phones produces a representation of sound that is sensible to many, but not all people. An underlying theme of the study is the importance of personal control for peripheral cues. Peripheral cues in the wild are better comprehended and less obtrusive if derived from music and are chosen by the intended recipient. Moreover, people have an overriding need to directly control the modality of cue delivery to manage etiquette. Context-adaptive cueing requires support and mechanisms for gaining a person's trust. Peripheral cues can provide a sparse overview of the underlying situation, but the ability to get details on demand is important to users, especially since the cues are ephemeral and sometimes not understood.

We conclude that despite the challenges presented by appropriating commodity sensors and actuators, that mobile phones are a suitable platform for proactive context awareness, at least for the "nice to know" case. Likewise, peripheral cues are a viable notification modality on mobile phones, despite their simple on/off actuators. Our current ongoing research is investigating the promise of other low-cost haptic communication techniques.

## 11. ACKNOWLEDGEMENTS

This was funded by UC Micro grants 06-186 and 07-067, Microsoft Research ER&P, a UCSD Chancellor's Fellowship, Motorola and supported in part by the UCSD FWGrid Project, NSF Research Infrastructure Grant number EIA-0303622. We also thank Intel Research Seattle for phones and data used in the analysis.

## 12. REFERENCES

[1] Berenzweig, A., Logan, B., Ellis, D. P. W, and Whitman, B., "A large-scale evaluation of acoustic and subjective music similarity measures", In *Proc of ISMIR*, 2003.

[2] Boff, K.R., Kaufman, L., Thomas, J.P., eds., Handbook of Perception and Human Performance: Sensory Processes and perception, vol. 2, John Wiley & Sons, 1986.

[3] Borovoy, R., Martin, F., Vemuri, S., Resnick, M., Silverman, B., and Hancock, C. Meme tags and community mirrors: moving from conferences to collaboration. In *Proc. of CSCW '98*, pp.159-168.

[4] Brewster, S. A. Using nonspeech sounds to provide navigation cues. *ACM TOCHI*, 224-259.

[5] Brewster, S. A. and Brown, L. M. 2004. Non-visual information display using tactons. *CHI '04 Extended Abstracts*.

[6] Brewster, S. A., Wright, P. C., and Edwards, A. D. An evaluation of earcons for use in auditory human-computer interfaces. In *Proceedings of the INTERACT '93 and CHI '93*, pp. 222-227.

[7] Brown, L. M., Brewster, S. A., and Purchase, H. C. Multidimensional tactons for non-visual information presentation in mobile devices. *Proc. of MobileHCI 2006*.

[8] Brown, L. M. and Kaaresoja, T. Feel who's talking: using tactons for mobile phone alerts. *CHI 2006 Extended.Abstracts*.

[9] Chang, A., O'Modhrain, S., Jacob, R., Gunther, E., and Ishii, H. ComTouch: design of a vibrotactile communication device. *Proc. of DIS '02*.

[10] Chang, A. and O'Sullivan, C. Audio-haptic feedback in mobile phones. In *CHI '05 Extended Abstracts, pp.* 1264-1267.

[11] Chen, M., Sohn, T., Chmelev, D., et al. Practical Metropolitan-Scale Positioning for GSM Phones. *In Proc of Ubicomp 2006*.

[12] Dodgeball. http://www.dodgeball.com

[13] Gaver, W.W. The SonicFinder: An interface that uses auditory icons. Human Computer Interaction, 4:67-94, 1989.

[14] Geldard, F.A., Adventures in tactile literacy. American Psychologist, 1957. 12: p.115-124.

[15] Geldard, F.A. Some Neglected Possibilities of Communication, Science, 131, 1960, 1583-1588.

[16] Griswold, W.G., Shanahan, P., Brown, S.W., Boyer, R.T. ActiveCampus: Experiments in Community-Oriented Ubiquitous Computing. IEEE Computer 37(10), 73-80.

[17] Hansson, R., Ljungstrand, P., Redström, J. Subtle and Public Notification Cues for Mobile Devices. *Proc. of UbiComp 2001*.

[18] Haykin, S., Van Veen, B. Signals and Systems. pp. 511-520. John Wiley & Sons, New York, CA. 1999.

[19] Ho, J. and Intille, S.S. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. *Proc. CHI '05*.

[20] Holmquist, L.E., J. Falk and J. Wigström (1999): Spporting Group Collaboration with Interpersonal Awareness Devices. Personal Technologies, vol. 3, pp. 13–21.

[21] Iachello, G., Smith., I., Consolvo, S., Abowd, G., Hughes, J., Howard, J., Potter, F., Scott, J., Sohn, T., Hightower, J., and LaMarca, A. Control, Deception, and Communication: Evaluating the Deployment of a Location-Enhanced Messaging Service. *Proc. of Ubicomp '05*.

[22] Isaacs, E., Walendowski, A., and Ranganthan, D. Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In Proc. of CHI '02 pp. 179-186.

[23] Immersion VibeTonz. http://www.immersion.com.

[24] Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interface Between People, Bits, and Atoms. *Proc. of CHI '97*, pp. 234-241.

[25] Krumm, J. and Hinckley, K. The NearMe Wireless Proximity Server. *Proc. of Ubicomp '04*.

[26] LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B. Place Lab: Device Positioning Using Radio Beacons in the Wild. *Proc. of Pervasive '05*.

[27] Loopt. http://www.loopt.com

[28] Luk, J., Pasquero, J., Little, S., MacLean, K., Levesque, V., and Hayward, V. 2006. A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. *Proc. of CHI '06*.

[29] Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. Heuristic evaluation of ambient displays. In Proc. of CHI '03.

[30] Marmasse, N., Schmandt, C., and Spectre, D. WatchMe: communication and awareness between members of a closely-knit group. *Proc. of Ubicomp '04*.

[31] Matthews, T., Dey, A. K., Mankoff, J., Carter, S., and Rattenbury, T. A toolkit for managing user attention in peripheral displays. In *Proc. of UIST '04* pp. 247-256.

[32] Mynatt, E. D., Back, M., Want, R., and Fredrick, R. Audio Aura: Light-Weight Audio Augmented Reality. *Proc. of UIST '97*.

[33] Patterson, D., Hennessey, J. Computer Organization and Design. pp501-02. Morgan Kaufmann, San Francisco, CA. 1998.

[34] Plazes. http://www.plazes.com

[35] Poupyrev, I., Maruyama, S., and Rekimoto, J. Ambient Touch: Designing tactile interfaces for handheld devices. *Proc. of UIST '02*.

[36] Sawhney, N. and Schmandt, C. Nomadic Radio: Scalable and Contextual Notification for Wearable Audio Messaging. *Proc. CHI '99*.

[37] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy of information visualizations, *Proc. IEEE Symposium on Visual Languages '96.*

[38] Smith, I., Consolvo, S., Hightower, J., Hughes, J., Iachello, G., LaMarca, A., Scott, J., Sohn, T., Abowd, G. Social Disclosure of Place: From Location Technology to Communication Practice. *Proc. of Pervasive 2005*.

[39] Sohn, T., Li, K.A., Lee, G., Smith, I., Scott, J., and Griswold W.G. Place-Its: A Study of Location-Based Reminders on Mobile Phones. *Proc. of Ubicomp 2005*.

[40] Sohn, T., Varshavsky, A., LaMarca, A., Chen, M.Y., Choudhury, T., Smith, I., Consolvo, S., Hightower, J., Griswold, W.G., and de Lara, E. Mobility Detection Using Everyday GSM Traces. *Proc. of Ubicomp 2006*.

[41] Tang, J. C., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., and Bhalodia, J. ConNexus to awarenex: extending awareness to mobile users. In *Proc of CHI '01* pp.221-228.

[42] Want, R., Hopper, A., Falcão, V., and Gibbons, J. The active badge location system. *ACM Trans. Inf. Syst.* 10, 1 (Jan. 1992), 91-102.

[43] Weiser, M. and Brown, J.S. Designing Calm Technology.

[44] http://www.ubiq.com/hypertext/weiser/calmtech/calmtech.htm.

[45] http://investing.reuters.co.uk/news/articleinvesting.aspx?type=media &storyID= nL291720