# Preprocessing in a Tiered Sensor Network for Habitat Monitoring

**Hanbiao Wang**

*Computer Science Department, University of California, Los Angeles (UCLA), Los Angeles, CA 90095-1596, USA*
*Email: hbwang@cs.ucla.edu*

**Deborah Estrin**

*Computer Science Department, University of California, Los Angeles (UCLA), Los Angeles, CA 90095-1596, USA*
*Email: destrin@cs.ucla.edu*

**Lewis Girod**

*Computer Science Department, University of California, Los Angeles (UCLA), Los Angeles, CA 90095-1596, USA*
*Email: girod@cs.ucla.edu*

We investigate task decomposition and collaboration in a two-tiered sensor network for habitat monitoring. The system recognizes and localizes a specified type of birdcalls. The system has a few powerful macronodes in the first tier, and many less powerful micronodes in the second tier. Each macronode combines data collected by multiple micronodes for target classification and localization. We describe two types of lightweight preprocessing which significantly reduce data transmission from micronodes to macronodes. Micronodes classify events according to their cross-zero rates and discard irrelevant events. Data about events of interest is reduced and compressed before being transmitted to macronodes for target localization. Preliminary experiments illustrate the effectiveness of event filtering and data reduction at micronodes.

**Keywords and phrases:** sensor network, collaborative signal processing, tiered architecture, classification, data reduction, data compression.

## 1. INTRODUCTION

Recent advances in wireless network, low-power circuit design, and micro electromechanical systems (MEMS) will enable pervasive sensing and will revolutionize the way in which we understand the physical world [1]. Extensive work has been done to address many aspects of wireless sensor network design, including low-power schemes [2, 3, 4], self-configuration [5], localization [6, 7, 8, 9, 10, 11], time synchronization [12, 13], data dissemination [14, 15, 16], and query processing [17]. This paper builds upon earlier work to address task decomposition and collaboration among nodes.

Although hardware for sensor network nodes will become smaller, cheaper, more powerful, and more energy-efficient, technological advances will never obviate the need to make trade-offs. Cerpa et al [18]. described a tiered hardware platform for habitat monitoring applications. Smaller, less capable nodes are used to exploit spatial diversity, while more powerful nodes combine and process the micronode sensing data.

Although details of task decomposition and collaboration clearly depend on the specific characteristics of applications, we hope to identify some common principles that can be applied to tiered sensor networks across various applications. We use birdcall recognition and localization as a case study of task decomposition and collaboration. In this context, we demonstrate two types of micronode preprocessing. Distributed detection algorithms and beamforming algorithms will not be discussed in detail in this paper although they are fundamental building blocks for our application.

The rest of the paper is organized as follows. Section 2 presents a two-tiered sensor network for habitat monitoring and the task decomposition and collaboration between tiers. Sections 3 and 4 illustrate two types of micronode preprocessing. Section 5 presents the preliminary results of data reduction and compression experiments. Section 6 is a brief description of related work. Section 7 concludes this paper.

## 2. TASK DECOMPOSITION AND COLLABORATION IN A TIERED SENSOR NETWORK FOR HABITAT MONITORING

### 2.1. Tiered sensor network for habitat monitoring

Our example application is the recognition and localization of a known acoustic source (e.g., a bird). The system first recognizes birdcalls of interest and then determines their locations.

Our two-tiered wireless sensor network is illustrated in Figure 1. It has two types of nodes: macronodes in the first tier and micronodes in the second tier. Micronodes are less expensive but more resource-constrained than macronodes. We choose commercial-off-the-shelf (CTOS) PC104 products as our macronodes http://www.pc104.org/consortium/. PC104 is a well-supported standard. They are physically small but available with CPUs ranging from i386 to Pentium II, memory up to 64 MB, and a full spectrum of peripheral devices including digital I/O, sensors and actuators. We choose the motes developed by UC Berkeley [19] and manufactured by Crossbow, Inc. as our micronodes http://www.xbow.com. The latest motes have 128-KB program memory, 4-KB data memory, 512-KB secondary storage, 50-Kb/s radio bandwidth, and 6 ADC channels. Both PC104s and motes can be equipped with acoustic sensors. Motes and PC104s can communicate with one another through wireless network. Micronodes can be densely distributed because of their low cost and small form factor. High density increases the probability for some micronodes to detect a stimulus close to its origin. Physical proximity to a stimulus yields higher SNR and improves opportunities for line of sight. Macronodes are sparsely distributed because of their higher power consumption. Nodes form a clustered wireless network by self-assembly [20]. Macronodes serve as cluster heads because they have more processing power and more capabilities than do micronodes. GPS on macronodes can provide location and time references to the rest of the system. Locations of other nodes can be determined iteratively, given a group of reference nodes' locations [6, 7, 10, 11]. Other nodes can also be synchronized to reference nodes [12, 13]. Figure 1 illustrates two clusters in a tiered sensor network.

### 2.2. Task decomposition and collaboration

The task of our case study system is to recognize the specified type of birdcalls and determine their locations. First, we need to specify the birdcalls of interest to the system as input. A convenient input format for biologists is the birdcall waveform. Biologists typically have recorded birdcall waveforms for the particular type of birds being studied. These waveforms can be input into the system from macronodes. The macronodes convert the waveforms into the internal formats used by birdcall recognition algorithms.

In particular, spectrograms are complete descriptions of bioacoustic characteristics of birdcalls. They are widely used by biologists for animal call classification. Macronodes have enough computational resources to use spectrograms internally to classify acoustic signals. However, micronodes are
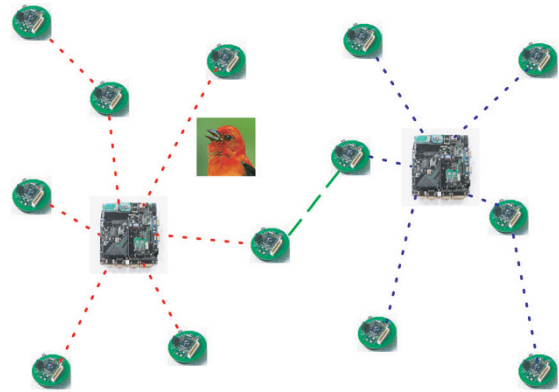


FIGURE 1: Two-tiered sensor network for bird monitoring. Macronodes are PC104s. Micronodes are Berkeley motes [19]. Dotted lines and dashed lines represent inner cluster and intercluster wireless communication links, respectively.

too resource-constrained to use spectrograms. We propose using a cross-zero rate representation for micronodes. Cross-zero rate is the rate at which a waveform changes signs. Consequently, this representation is always two times the most significant frequency and thus a summary of the most significant characteristics of a waveform. Figure 2 illustrates the relationship between spectrograms and cross-zero rates in Section 3. Cross-zero rates are easy to compute and easy to use. Classification using cross-zero rates will be discussed in detail in Section 3.

The target recognition task can be divided into two steps. All nodes first independently determine whether their acoustic signals are of the specified type of birdcalls. Then, macronodes can fuse all individual decisions into a more reliable system-level decision using distributed detection algorithms [21]. We will not discuss details of the decision fusion in this paper. We will describe how individual decisions are made in detail in Section 3.

The target localization task can also be divided into two steps. First, waveforms are recorded at nodes that are distributed at different locations. Second, all those data are accumulated to one macro node, and beamforming is applied to determine the target location. The procedure of the beamforming estimates target location using the time difference of arrival (TDOA) from a set of distributed sensors whose locations are known [22, 23, 24]. The time lag of the cross-correlation maximum between waveforms of the same target from two different sensors indicates TDOA between those two sensors.

So far, we have decomposed tasks and distributed them to appropriate nodes in order to optimize the cost effectiveness. Micronodes are densely distributed for sensing while macronodes are sparsely distributed for time-space reference and information fusion. Such optimization is one of the fundamental goals of task decomposition and collaboration in a tiered sensor network. However, there are also secondary goals that can significantly contribute to a longer lifetime for the system. For example, communication among
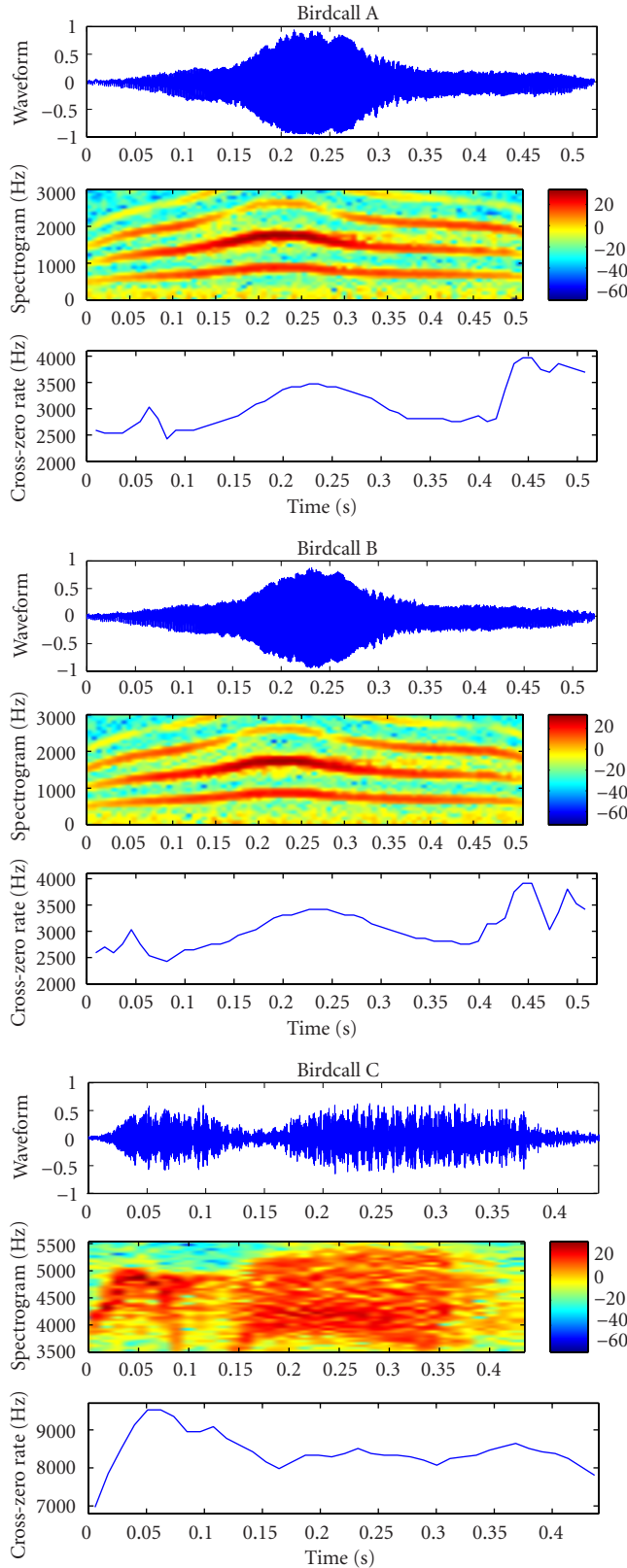
FIGURE 2: Waveforms, spectrograms, and cross-zero rates of birdcalls A, B, and C. Birdcalls A and B are of the same type while birdcall C is different. Spectrograms are only shown in a limited frequency band. The cross-zero rates are calculated in a time window of 20 ms.

nodes should be minimized because it is the primary energy consumer. Pottie and Kaiser have pointed out in [25] that each bit transmitted on the air will bring the node battery one step closer to its death. In the rest of this paper, we will discuss in detail two types of preprocessing at micronodes, which significantly reduce the data transmission overhead.

The first type of preprocessing is to recognize events of interest and filter out irrelevant events at the micronodes. When waveforms of a specific type of birdcalls are input to the system at a macro node, the macro node computes its spectrogram and cross-zero rate and sends the spectrogram and the cross-zero rate to all other macronodes. All macronodes broadcast the cross-zero rate to all micronodes in their respective clusters. Micronodes use the cross-zero rate to determine whether a detected signal is of the specified type of birdcalls or not. If it is not, it will be discarded without being further sent to its cluster head for data fusion. Assuming events of interest occur sparsely in the long lifetime of a sensor network, the local filtering at micronodes will significantly reduce the amount of data that needs to be transmitted to macronodes.

The second type of preprocessing is to do data reduction/compression at the sensor nodes before data is transmitted to the macro node for combination. Data reduction reduces data size by discarding irrelevant information in data.[1] In our example of sensor network, source location estimation needs arrival-time information of acoustic signals at multiple sensor nodes. We use an audio reduction/compression technique that retains most time information in audio waveforms while discarding amplitude change details. Cross correlation between two waveforms of the same stimulus recorded at two different locations indicates TDOA between those two locations. Cross correlation of two reduced/compressed waveforms indicates the same TDOA as the cross correlation of their respective raw waveforms does.

The above two components have the potential to greatly reduce the amount of wireless communication and energy cost in the sensor network. As a result, the system lifetime will be extended. The remainder of this paper describes specific techniques to implement these two types of processing at micronodes.

## 3. EVENT FILTERING AT MICRONODES

We now describe the first type of preprocessing at micronodes—a lightweight event recognition scheme that identifies events of interest while discarding irrelevant events. In our case study of bird monitoring application, motes will be exposed to acoustic signals from all kinds of events such as wind, rain, traffic, and other animal calls. We use micron-

---

[1] The semantics of irrelevant information is determined by the characteristics of the application. For example, MP3 compression uses the *psychoacoustic* selection of sound signals to eliminate those signals that we are unable to hear while retaining human perception. Therefore, sounds below the minimum audition threshold and sounds masked by stronger sounds are irrelevant information.

odes to determine event type locally and discard signals of irrelevant events.

The traditional birdcall classification is based on bioacoustics. Spectrograms completely describe bioacoustic characteristics of each type of birdcalls. When the spectrogram is computed for an observed acoustic signal, any standard detection methods for two-dimensional signals can be applied to determine whether the spectrogram is of the type of birdcalls of interest or not. One of the straightforward classification methods uses the cross-correlation coefficient between the measured spectrogram and the reference spectrogram. In Figure 2, there are three birdcalls. Birdcalls A and B are of the same type, and their cross-correlation coefficient is about 97%. Birdcalls A and C are of different types, and their cross-correlation coefficient is 0%. We can choose a threshold for cross-correlation coefficients. All cross-correlation coefficients beyond the threshold indicate that two birdcalls are of the same type.

Computation of spectrograms and cross-correlation coefficients demands much CPU and memory. For example, it takes our macro node of 266 MHz CPU and 64 MB RAM more than 300 ms to complete a classification operation using the cross-correlation coefficient between the measured spectrogram and the reference spectrogram. As described earlier, we thus use the cross-zero rate of the detected signal to determine its event type. When signal samples stream into the micronode, the cross-zero rate can be easily computed by simply counting the number of zero-crossings, which demands much less computational resource than the spectrogram. One of the straightforward classification methods using cross-zero rates is to use the average difference of two cross-zero rate curves. In Figure 2, the same type of birdcalls A and B have an average cross-zero rate difference of 84 Hz while different types of birdcalls A and C have an average cross-zero rate difference of 5416 Hz. Computation of the average difference between two cross-zero rate curves also costs much less resource than computation of cross-correlation coefficient between two spectrograms. We choose a threshold for the average difference between two cross-zero rate curves. An average difference between two cross-zero rate curves below the threshold indicates that the two birdcalls are of the same type.

The advantage of cross-zero rates comes from its low computational resource demands. However, the cross-zero rate loses some information about the spectrogram. When noise is so strong that the most significant frequency is from noise instead of a birdcall, the cross-zero rate will be distorted. The distorted cross-zero rate curve represents characteristics of noise, not of the birdcall. When noise is not strong enough to change the most significant frequency in data, noise has no effect on the cross-zero rate at all because the cross-zero rate is only determined by the most significant frequency in data. Fortunately, birdcalls usually have a narrow bandwidth. Therefore, we can filter out the noise that is not in the bandwidth of the birdcall to be monitored. For example, the noise caused by wind in the outdoor environment usually has much lower frequency than typical birdcalls. Therefore, wind can be easily filtered out. Filtering

is the first stage of processing after signals are sampled at micronodes. The computational cost of simple bandpass filtering is low enough for micronodes to handle. However, when noise is in the same bandwidth as the birdcalls to be monitored, filtering does not help. For example, a birdcall of interest could be so severely polluted by other animal calls that the measured cross-zero rate curve does not match the reference cross-zero rate curve. In that scenario, birdcalls of the specified type indeed could be discarded as irrelevant calls. In rare cases, two different types of acoustic signals may have similar cross-zero rates although their spectrograms are different.

## 4. DATA REDUCTION/COMPRESSION AT MICRONODES

In this section, we describe the second type of preprocessing at micronodes, a data reduction scheme that retains most time information of acoustic signals for beamforming using TDOA. We also present S-coding that compactly encodes reduced acoustic signals. After reduction and compression, data will be sent to macronodes.

### 4.1. Data reduction

In the example of sensor network for bird monitoring, the source location estimation requires beamforming of signals detected by multiple micronodes. The simplest design is for all micronodes to send all the waveforms to a macro node for beamforming. However, the bandwidth and energy consumption are far beyond the capability of the system. A sampling rate of 22 kHz with a sample size of 8 bits will generate data at a rate higher than three times of what a micronode's 50-Kbps radio can transmit. Moreover, the energy consumption would greatly shorten the system lifetime. Instead, micronodes must reduce/compress raw data locally before it is sent to the macro node.

Data reduction based on application characteristics is not a new concept. In estimation theory, minimum sufficient statistics is a function of a set of samples [26]. It contains no less information about the parameter to be estimated than the original set of samples while having much smaller data size. This concept can also be generalized to apply to signal processing in sensor network. The following describes a specific data reduction scheme used in our case study of sensor network. It transforms raw waveforms into a coarse format with smaller data size while keeping most time information contained in raw waveforms. Specifically, the cross-correlation of reduced waveforms indicates the same TDOA as raw waveforms. Thus, TDOA-based beamforming can use reduced waveforms instead of raw waveforms to determine the target location. TDOA-based beamforming has been discussed in detail in many papers [22, 23, 24].

A typical digitized raw signal waveform is a sequence of real-valued signal samples, where indices indicate the time,

$$\{a_i \mid i = 0, \ldots, n - 1\}. \tag{1}$$

We define a *segment* as a consecutive subsequence of the waveform, within which all samples have the same signs,

but immediately-before or immediately-after samples have different signs. For any physical signal sampled at proper rate, $\{a_i\}$ is actually a sequence of alternate positive-signed segments and negative-signed segments. Our data reduction scheme for a waveform is based on the following important observation.[2] Most of the time information of the waveform is contained in the moments when alternate transitions between positive-signed segments and negative-signed segments occur. The signal variation details within a segment can be discarded with little loss of time information. The following coarse waveform $\{b_i\}$ contains most of the time information contained in the raw waveform $\{a_i\}$:

$$\{b_i \mid i = 0, \ldots, n - 1\}, \tag{2}$$

where

$$b_i = \begin{cases} +1, & \text{if } a_i \geq 0, \\ -1, & \text{if } a_i < 0. \end{cases} \tag{3}$$

Therefore, $\{b_i\}$ can replace $\{a_i\}$ without causing much loss of time information.

After micronodes reduced the raw waveform $\{a_i\}$ into the coarse waveform $\{b_i\}$, there are two options. One is to code $\{b_i\}$ into a binary string (+1 encoded as 1 and −1 encoded as 0) before sending it to macronodes. When the raw waveform has a sample size of $n$ bits, then the total size of the reduced waveform is only $1/n$ of the total size of the raw waveform. The second option is to view the coarse waveform $\{b_i\}$ as a sequence of segments, which can be completely represented by the sign of the first segment, the starting time of the first segment, and a *sequence of segment lengths (SSL)*. The SSL representation can be further encoded into a more compact format. In either case, data reduction can significantly reduce data transmission by reducing raw waveforms into course waveforms. Motivated by bigger compress gains, we will discuss the second option in detail in the following paragraphs.

We have discussed the effects of noise to cross-zero rate in Section 3. When noise is strong enough to alter the most significant frequency component of the data to be classified, noise must be filtered out before computing cross-zero rate. Otherwise, cross-zero rate will represent characteristics of noise instead of the birdcall to be classified. Likewise, strong noise must also be filtered before data reduction. Otherwise, the coarse waveform will represent the time information of noise arrival at sensors. Fortunately, the noise is low enough in birdcalls that have already been classified as the type of interest using cross-zero rate. Otherwise, classification using cross-zero rate will discard the birdcall as irrelevant events. Thus, data reduction applied after classification using cross-zero rate is safe from noise corruption and thus retain the right time information of signals. Therefore, filtering is crit-

---
[2]We were inspired by personal communication with Dr. Ralph Hudson and Dr. Kung Yao. Dr. Hudson and Dr. Yao suggested that cross correlation between waveforms sampled at extreme sample size of 1 bit still indicates the correct TDOA.

TABLE 1: Base-16 *S*-code.

| Number range | Base-16 *S*-code |
| --- | --- |
| 1, 15 | 0x 1, 0x F |
| 16, 255 | 0x 0 10, 0x 0 FF |
| 256, 4095 | 0x 00 100, 0x 00 FFF |

ical to both cross-zero rate-based classification and data reduction when noise is strong. In order to make cross-zero rate-based classification and data reduction valid, the first step of preprocessing immediately after sampling should be noise filtering.

### 4.2. Data encoding

The sign and starting time of the first segment can be efficiently encoded in a constant amount of space. However, depending on segment length distribution, it takes variable space to encode an SSL. For convenience, we will not differentiate terms for the whole encoding task and the encoding of its SSL.

An SSL is a sequence of natural numbers in which most segments have a few samples while a few segments could have many samples. To encode an SSL is a problem of variable-length coding of natural numbers. Many variable-length coding of integers have been proposed [27, 28, 29, 30, 31, 32]. However, there is no "best" encoding scheme because encoding efficiency always depends on the probability distribution of integers to be encoded. Many encoding schemes may be able to encode SSL with high efficiency. For convenience, we propose to use *S*-code for the encoding of SSL. *S*-code is an extension of Elias $\gamma'$-code [28, 29]. Elias $\gamma'$-code usually consists of two parts: flag bits and data bits. Flag bits tell how many data bits are used for the number. It produces shorter codes for small integers and longer codes for large integers. Unlike Elias $\gamma'$-code which is binary number, *S*-code is base-$2^N$ number instead. Like Elias $\gamma'$-code, *S*-code is the concatenation of flag bits and data bits. Flag bits indicate coding length of the integer. Elias $\gamma'$-code has no flag bit for 1. Likewise, *S*-code has no flag bits for natural number smaller than $2^N$. Data bits are simply direct unsigned representation of the natural number. When $N = 1$, *S*-code turns into Elias $\gamma'$-code. Table 1 shows base-$2^4$ (hexadecimal) *S*-code.

Because sampling rate is often several times the cutoff frequency of signals, the shortest segment has several samples. Because birdcalls are usually limited in a narrow bandwidth from tens of Hz to several kHz, length of the longest segment will be no longer than 100 times of that of the shortest segment. Each type of birdcalls has its characteristic segment length distribution for a given sampling rate. Given the segment length distribution and base $2^N$ used for *S*-code, the size of *S*-coded SSL can be analytically predicted. To maximize compression efficiency of *S*-code, this $N$ should be chosen such that most segment lengths are between $2^N - 1$ and $2^N$. Because the encoding size can be predicted when the event type of interest is specified to the sensor network, we can specify the optimal value of $N$ before sensor nodes start data compression.

After an SSL is *S*-coded, general purpose compression such as zip can be applied in addition. Our preliminary experiments show that both encoding methods have significant compression gain.

## 5. EXPERIMENTS

The purpose of our experiments is to explore the validity and efficiency of the proposed data reduction and compression schemes. In our experiments, a birdcall is recorded with two synchronized microphones. The cross correlation between waveforms of those two channels indicates TDOA between two microphones. We apply our data reduction/compression to the raw waveforms as in (1) and then decode it into a coarse waveform as in (2). The cross correlation between coarse waveforms indicates almost the same TDOA as that between the corresponding raw waveforms. The error is within one sample interval. Therefore, the data reduction scheme appears to retain most time information in raw waveforms. When data reduction, *S*-coding, and zipping are applied to raw waveforms in order, the overall compression ratio is 69.6 on average.

### 5.1. Experiment method

The experiments were done in an outdoor environment with noise of traffic and venting. Temperature, humidity, and wind speed are 55 F, 49%, and 12 mph, respectively. Estimated sound speed was approximately 339.5 m/s, based on the algorithm in [33]. The birdcall was played back from a standard computer speaker driven by an Compaq iPAQ pocket PC H3760. Sound was recorded with a pair of synchronized microphones connected to a laptop. Sampling rate is 32 kHz. Sample size is 16 bits. Both speaker and microphones were mounted above ground 6 feet and in one straight line. Two microphones were separated by approximately 9 feet.

There are two groups of recording experiments. In the first group of experiments, the speaker was put at four different positions, as Figure 3 shows, with the same volume. In the second group of experiments, the speaker was turned to four different volumes at the same position as $S_1$ in Figure 3 indicates.

### 5.2. Recorded waveforms

Figure 4 shows recorded waveforms in the first group of experiments. Figure 5 shows recorded waveforms in the second group of experiments. $S_1$ and $V_1$ are the same recording experiment. They are put into two groups for purpose of comparison.

### 5.3. Validity of data reduction/compression

We applied data reduction/compression to recorded waveforms and then restored coarse waveforms from the encoding. TDOA was computed using cross correlation between two coarse waveforms. For comparison, we also computed TDOA using cross correlation of raw waveforms. TDOA between L and R channels are listed in Table 2 in unit of sam-
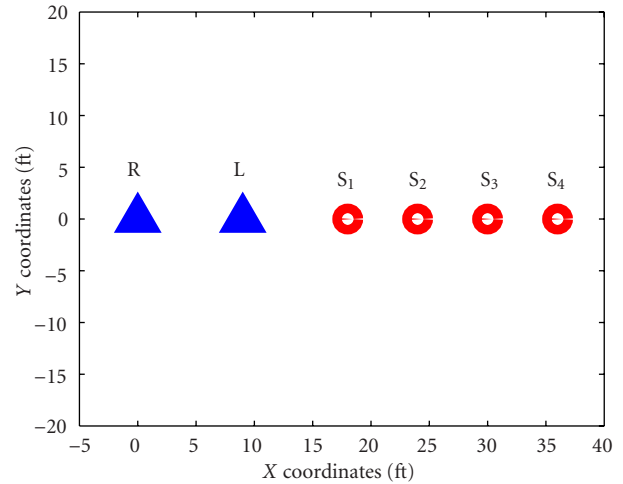


FIGURE 3: Microphones and speaker positions. Microphones are located at the triangles and speakers are located at circles. L and R are the left and right channels of the synchronized microphones pairs. $S_1$, $S_2$, $S_3$, and $S_4$ are four positions of the speaker.

ple intervals (1/32000 second). TDOA computed from raw waveforms are 261 sample intervals. Given sampling rate 32 kHz and sound speed estimation 339.5 m/s, TDOA corresponds to 261/32000*339.5 m/s = 2.769 m, which is consistent with the distance between two microphones. TDOA computed from coarse waveforms are within ±1 sample interval from TDOA indicated by raw waveforms. Our data reduction essentially keeps all positions of zero crossings in the recorded raw waveform. Because the resolution of cross-zero position is one sample interval, it is reasonable to see error of ±1 sample interval in TDOA indicated by coarse waveforms. Therefore, our data reduction appears to retain almost all time information in the raw waveforms. Figure 6 shows cross correlation between L/R coarse waveforms of $S_1$.

### 5.4. Efficiency of data reduction/compression

Table 3 shows data size of waveforms and their reduced/*S*-coded/zipped formats. Data size of all raw waveforms is $16,000 \times 16 = 256,000$ bits. Data reduction reduces a raw waveform as in (1) to a coarse waveform as in (2). A coarse waveform is completely represented by the sign and the starting time of the first segment and SSL. Because SSL takes more than 99% space of coarse waveform representation, we will not differentiate SSL and coarse waveform representation for purpose of compression ratio analysis. No segment has more than 65,535 samples. Therefore, Each segment length can be represented by a 16-bit natural number in SSL. Reduction efficiency is given by the ratio of raw waveform size to SSL size. The average reduction efficiency is about 11.4.

*S*-coding encodes SSL into a compact format. Base-16 *S*-coding is chosen because most segment lengths are between 8 and 16. A typical probability distribution of segment lengths is shown in Figure 7. Efficiency of *S*-coding is the ratio of SSL size to the size of *S*-coded SSL. The average *S*-coding efficiency is about 3.3. In order to compare the performance of
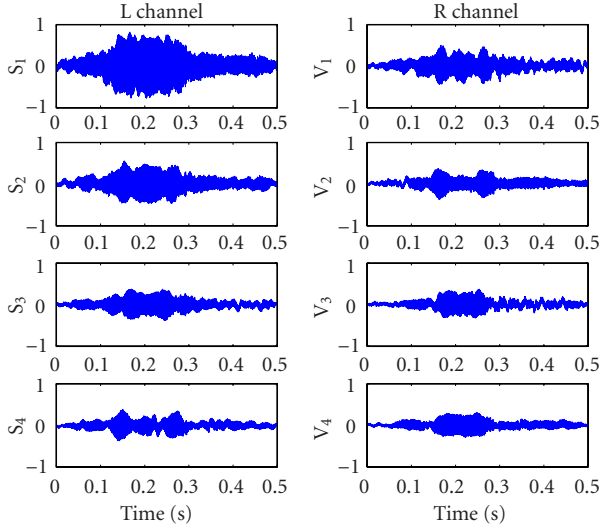
FIGURE 4: Recorded waveforms by a pair of synchronized microphones. Microphone speaker positions are shown in Figure 3. In the above four recording experiments, the speaker volume is the same while the distances from the speaker to the pair of microphones are different.
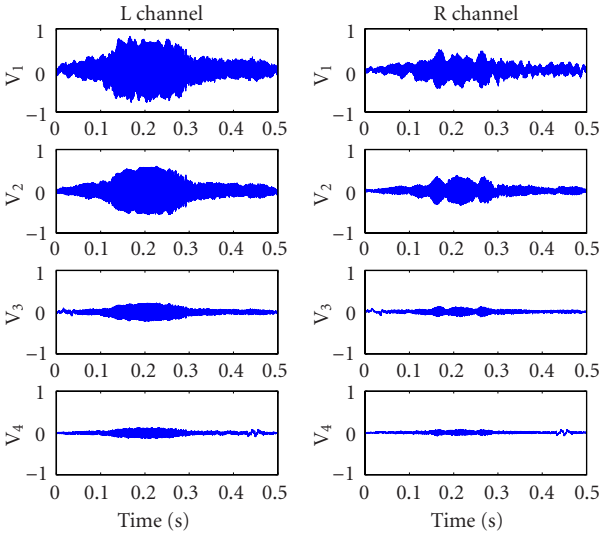


FIGURE 5: Recorded waveforms by a pair of synchronized microphones. Microphone geometry is shown in Figure 3. The speaker is located at $S_1$ in Figure 3. In the above four recording experiments, the speaker volumes are in a decreasing order from $V_1$ to $V_4$ while the distances from the speaker to the pair of microphones are the same.

$S$-coding to that of general-purpose compression algorithms, we compress SSL with WinZip 8.0. Zipping efficiency is the ratio of SSL size to the size of zipped SSL. The average zipping efficiency is about 2.7.

We also examine the efficiency of $S$-coding followed by zipping. It is the ratio of SSL size to the size of zipped $S$-coded SSL. The average efficiency of concatenation of $S$-coding and zipping is about 6.1. It is significantly larger than that of

TABLE 2: TDOA indicated by cross correlation of raw waveforms and of coarse waveforms.

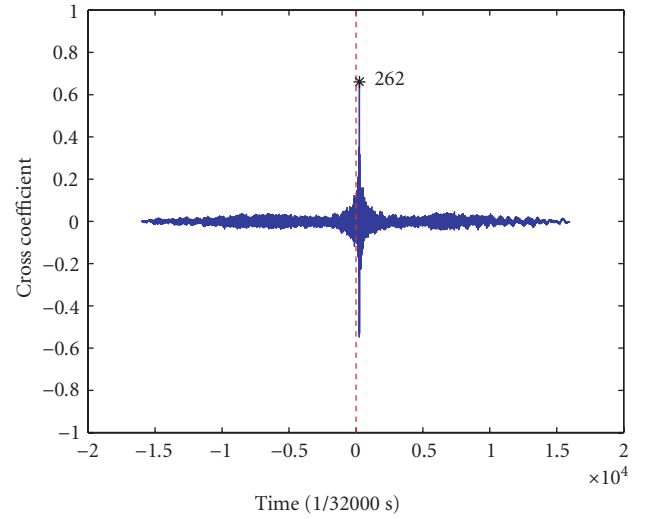| Record | TDOA from raw waveforms (sample interval) | TDOA from coarse waveforms (sample interval) |
|---|---|---|
| $S_1/V_1$ | 261 | 261 |
| $S_2$ | 261 | 261 |
| $S_3$ | 261 | 261 |
| $S_4$ | 261 | 261 |
| $V_2$ | 261 | 262 |
| $V_3$ | 261 | 260 |
| $V_4$ | 261 | 260 |



FIGURE 6: Cross coefficient between coarse waveforms of $S_1$ indicates TDOA of 262 sample intervals. Dashed line represents TDOA of 0. The star indicates the peak of the cross coefficient, which has an offset of 262 sample intervals from dashed line.

$S$-coding or zipping if applied individually. It indicates that $S$-coding and zipping are somewhat orthogonal to each other. They exploit different redundancy in SSL. Therefore, it is possible to design a more sophisticated compression algorithm that combines the power of both $S$-coding and zipping. However, $S$-coding is quite simple and good for low-end micronodes such as motes. When the sensor nodes have enough processing capability to run a more sophisticated compression algorithm than $S$-coding, we may just apply $S$-coding followed by zipping.

When data reduction, $S$-coding, and zipping are applied in order, the ratio of raw waveform size to the size of zipped $S$-coded SSL is 69.6, which is much larger than that of existing data compression schemes for audio data.

## 6.  RELATED WORK AND DISCUSSION

Pottie [25, 34] pointed out that subnetworks should be formed in a large wireless sensor network. The subnetwork

TABLE 3: Data size of reduced/$S$-coded/zipped waveforms (all raw waveforms have 256,000 bits).

| Record | SSL (bit) | Zipped SSL (bit) | $S$-coded SSL (bit) | Zipped $S$-coded SSL (bit) |
|---|---|---|---|---|
| $S_1/V_1$ (L) | 24,048 | 7,544 | 6,944 | 3,064 |
| $S_1/V_1$ (R) | 23,120 | 8,664 | 6,784 | 3,832 |
| $S_2$ (L) | 23,696 | 7,992 | 6,928 | 3,200 |
| $S_2$ (R) | 23,088 | 8,440 | 6,832 | 2,032 |
| $S_3$ (L) | 23,040 | 8,424 | 6,780 | 3,464 |
| $S_3$ (R) | 22,400 | 8,616 | 6,628 | 3,832 |
| $S_4$ (L) | 21,472 | 8,816 | 6,852 | 4,560 |
| $S_4$ (R) | 22,016 | 8,984 | 6,900 | 4,008 |
| $V_2$ (L) | 23,728 | 7,048 | 6,904 | 2,784 |
| $V_2$ (R) | 23,968 | 8,168 | 7,020 | 3,904 |
| $V_3$ (L) | 23,248 | 7,664 | 6,872 | 3,192 |
| $V_3$ (R) | 21,618 | 8,728 | 6,424 | 3,328 |
| $V_4$ (L) | 21,472 | 8,536 | 6,816 | 3,632 |
| $V_4$ (R) | 17,504 | 8,296 | 5,804 | 4,648 |
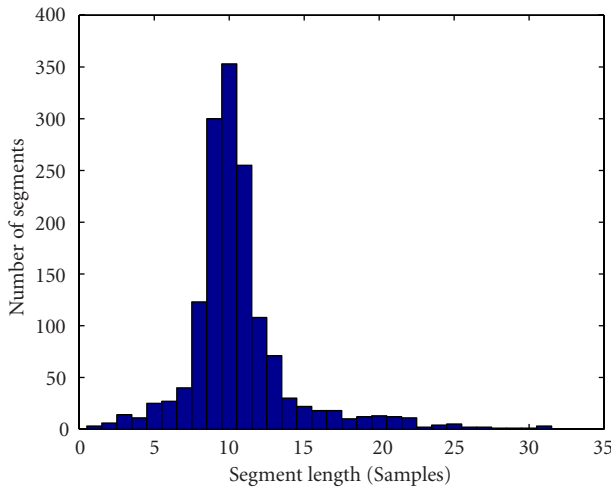


FIGURE 7: Probability distribution of segment lengths for $S_1$. Because most segment lengths are between 8 ($2^3$) and 16 ($2^4$), base-16 $S$-coding has the maximum compression gain.

organization enables coordinated internal communication by a master so that some internal nodes can be powered down. Many possible trade-offs related to architecture of wireless sensor network were also extensively discussed in [34]. He concluded that the high cost of wireless communication compared to data processing leads to a different trade-off regime other than that of traditional ad hoc wireless network. The trade-off between homogeneous and heterogeneous nodes is briefly discussed. However, there were no detailed discussions on task decomposition and collaboration in a tiered architecture, especially preprocessing at micronodes.

Van Dych and Miller [35] proposed a cluster-based architecture for sensor networks motivated by the performance of distributed detection algorithms. However, there is a significant difference between their focus and ours. They focus on the scenario of distributed sensing and detection. Binary decisions are made at local sensing nodes and there is no need for transmission of raw signals. We focus on coherent signal processing scenarios that have much higher demands on bandwidth than distributed detections. We choose the hierarchical organization of sensor networks in order to reduce wireless communication and thus energy consumption by distributing signal processing to local micronodes and clusters. For coherent signal processing, either raw signal or its reduced format must be collected to a central node for information fusion. We propose a data reduction scheme at micronodes for acoustic signals. However, there is no need for such data reduction scheme in the distributed detection scenario in [35].

Tiered sensor network hardware platforms were proposed by Cerpa et al. [18] for habitat monitoring applications. They pointed out that larger, faster, and more expensive hardware can be used more effectively together with small factor nodes because the later can be densely distributed and have small form factor. However, software architecture or task decomposition and collaboration mechanisms for in-network signal processing was not addressed for the tiered architecture in [18].

Mainwaring et al. [36] also describe a tiered sensor network for habitat monitoring on Great Duck Island (GDI). Their application monitors environment conditions such as light, temperature, barometric pressure, humidity, and infrared. They use a tiered architecture solely for communication. The lowest level consists of sensor nodes deployed in dense patches that could be widely separated. In each sensor patch, a gateway node transmits data from the patch to a base station that serves the collection of patches. The base station transmits all data to a central database through the Internet. In contrast, we propose a tiered

architecture for the purposes of collaborative signal and information processing inside the sensor network. We deploy a hierarchy of nodes to accommodate demanding data processing tasks that cannot be handled by smaller sensor nodes. The GDI system described does not require collaborative data processing inside their sensor network. All data is transmitted back to a central database for off-line data mining and analysis. It is feasible to transmit data sampled at those relatively low rates all the way back without local processing. However, in our application context, it is not feasible to transmit all the data back due to the higher sampling rate. For a network of 1000 sensor nodes that sample acoustic signal at 20 kHz with a sample size of 16 bits, the data generation rate is 320 Mbps, which is infeasible with the existing wireless network technology on nodes of small form factor and constrained-energy resource. We propose in-network processing of birdcalls to generate high-level descriptions such as birdcall type, calling time, and location. Then, the high-level description of smaller data size can be transmitted back for further analysis by biologists. In summary, the Mainwaring et al. system, the birdcall recognition, and the localization system described here are largely complementary.

## 7. CONCLUSION

Minimization of communication is a principle goal of task decomposition and collaboration in tiered sensor networks due to energy constraints. We describe local filtering and data reduction as two types of preprocessing at micronodes that significantly reduce data transmission to macronodes. This paper presents only preliminary experimental evidence which shows that both data reduction and event filtering using cross-zero rate are valid and effective. Future work must include construction and evaluation of a complete system.

### ACKNOWLEDGMENTS

### REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 70–84, Rome, Italy, July 2001.

[3] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," Tech. Rep. ISI-TR-543, USC/Information Sciences Institute, University of Southern California, Los Angeles, Calif, USA, September 2001.

[4] J. M. Rabaey, M. J. Ammer, J. L. Da Silva Jr., D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking," *IEEE Computer Magazine*, vol. 33, no. 7, pp. 42–48, 2000.

[5] A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor network topologies," in *Proc. 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom '02)*, New York, NY, USA, June 2002.

[6] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: self-configuring localization systems," in *Proc. 6th International Symposium on Communication Theory and Applications (ISCTA '01)*, Ambleside, Lake District, UK, July 2001.

[7] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, USA, October 2001.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 32–43, Boston, Mass, USA, August 2000.

[9] N. B. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," in *Proc. 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 1–14, Rome, Italy, July 2001.

[10] L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin, "Locating tiny sensors in time and space: A case study," in *Proc. IEEE International Conference on Computer Design*, Freiburg, Germany, September 2002.

[11] A. Savvides, C.-C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 166–179, Rome, Italy, July 2001.

[12] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in *Proc. 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing*, p. 186, San Francisco, Calif, USA, April 2001.

[13] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, Mass, USA, December 2002.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 56–67, Boston, Mass, USA, August 2000.

[15] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal on High Performance Computing Applications*, vol. 16, no. 3, pp. 293–314, 2002.

[16] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 174–185, Seattle, Wash, USA, August 1999.

[17] P. Bonnet, J. E. Gehrke, and P. Seshadri, "Querying the physical world," *IEEE Personal Communications*, vol. 7, no. 5, pp. 10–15, 2000, Special Issue on Smart Spaces and Environments.

[18] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," in *Proc. ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, Costa Rica, April 2001.

[19] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart dust: communicating with a cubic-millimeter computer," *IEEE Computer Magazine*, vol. 34, no. 1, pp. 44–51, 2001.

[20] K. Sohrabi, W. Merrill, J. Elson, L. Girod, F. Newberg, and W. Kaiser, "Scaleable self-assembly for ad hoc wireless sensor networks," in *Proc. IEEE CAS Workshop on Wireless Communications and Networking*, Pasadena, Calif, USA, September 2002.

[21] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part I-fundamentals," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 54–63, 1997.

[22] C. W. Reed, R. Hudson, and K. Yao, "Direct joint Source localization and propagation speed estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, pp. 1169–1172, Phoenix, Ariz, USA, March 1999.

[23] T. L. Tung, K. Yao, C. W. Reed, R. E. Hudson, D. Chen, and J. C. Chen, "Source localization and time delay estimation using constrained least squares and best path smoothing," in *Advanced Signal Processing Algorithms, Architectures, and Implementations IX*, vol. 3807 of *SPIE Proceedings*, pp. 220–233, Los Angeles, Calif, USA, July 1999.

[24] H. Wang, L. Yip, D. Maniezzo, et al., "A wireless time-synchronized COTS sensor platform: applications to beamforming," in *Proc. IEEE CAS Workshop on Wireless Communication and Networking*, Pasadena, Calif, USA, September 2002.

[25] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[26] R. N. McDonough and A. D. Whalen, *Detection of Signals in Noise*, Academic Press, Orlando, Fla, USA, 1995.

[27] S. W. Golomb, "Run-length encodings," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, 1966.

[28] V. E. Levenstein, "On the redundancy and delay of separable codes for the natural numbers," *Problems of Cybernetics*, vol. 20, pp. 173–179, 1968.

[29] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. IT-21, no. 2, pp. 194–203, 1975.

[30] R. F. Rice, *Some Practical Universal Noiseless Coding Techniques*, vol. 79-22 of *JPL Publication*, Jet Propulsion Laboratory, Pasadena, Calif, USA, 1979.

[31] E. R. Fiala and D. H. Greene, "Data compression with finite windows," *Communications of the ACM*, vol. 32, no. 4, pp. 490–505, 1989.

[32] P. Fenwich, "Punctured Elias codes for variable-length coding of the integers," Tech. Rep. 137, Department of Computer Science, The University of Auckland, Auckland, New Zealand, December 1996.

[33] O. Cramer, "The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and CO2 concentration," *Journal of the Acoustical Society of America*, vol. 93, pp. 2510–2516, May 1993.

[34] G. J. Pottie, "Wireless sensor networks," in *Proc. IEEE Information Theory Workshop*, pp. 139–140, Killarney, Ireland, June 1998.

[35] R. E. Van Dyck and L. E. Miller, "Distributed sensor processing over an ad hoc wireless network: simulation framework and performance criteria," in *Proc. MILCOM*, Washington, DC, USA, October 2001.

[36] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, Atlanta, Ga, USA, September 2002.

**Hanbiao Wang** is a third-year Ph.D. student of computer science at UCLA. He is currently working on collaborative information and signal processing in sensor networks. He is very interested in designing energy and bandwidth-efficient sensor networks by intertwining tasks of networking and information processing. He received his B.S. degree in geophysics from University of Science and Technology of China. He also received an M.S. degree in geophysics and space physics, and an M.S. degree in computer science from UCLA. He is a member of the ACM and the IEEE.
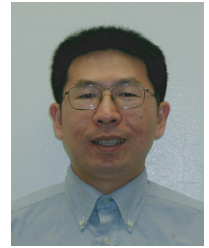
**Deborah Estrin** is a Professor of computer science at UCLA and Director of the Center for Embedded Networked Sensing (CENS), a newly awarded National Science Foundation Science and Technology Center. She received her Ph.D. degree in computer science from MIT (1985) and was on the faculty of Computer Science at USC from 1986 till mid-2000 where she received the National Science Foundation, Presidential Young Investigator Award for her research in network interconnection and security (1987). During the subsequent 10 years, her research focused on the design of network and routing protocols for very large global networks. Estrin has been instrumental in defining the national research agenda for wireless sensor networks, first chairing a 1998 DARPA ISAT study and then a 2001 NRC study; the latter culminated in an NRC publication—*Embedded Everywhere: A Research Agenda for Networked System of Embedded Computers*. Estrin's research group develops algorithms and systems to support rapidly deployable and robustly operating networks of many thousands of physically-embedded devices. She is particularly interested in applications to environmental monitoring. Estrin has served on numerous program committees and editorial boards, including SIGCOMM, Mobicom, SOSP, and ACM/IEEE Transactions on Networks. She is a Fellow of the ACM and AAAS.

**Lewis Girod** received his B.S. and M.E. in computer science from MIT in 1995. After working at LCS for two years in the area of Internet naming infrastructure, he joined Deborah Estrin's group as a Ph.D. student in 1998. He is currently a Ph.D. candidate at UCLA. His research focus is the development of robust networked sensor systems, specifically physical localization systems that use multiple sensor modalities to operate independently of environment and deployment.