Recognition of Complex Settings by Aggregating Atomic Scenes

Waltenegus Dargie and Tobias Tersch, *Technical University of Dresden*

This approach imitates human reasoning to enable flexible context recognition. Its usefulness is demonstrated by employing audio-signal processing to recognize several everyday situations.

One important aspect of ubiquitous computing is context awareness, which aims to establish a shared understanding of the user's social and conceptual settings (contexts). Establishing such a shared understanding can be simple or complex. By simple, we mean that you can easily obtain the necessary sensors and can map the sensed data to a meaningful setting. In most cases, however, the process is complex and the end result is uncertain. In the latter situation, context acquisition involves modeling and reasoning about the characteristics of and relationships between several entities.

Most approaches to context reasoning model complex settings (higher-level contexts) as monolithic scenes rather than aggregations of distinct scenes. For example, recognition of a street setting on the basis of features extracted from an audio signal requires an existing model of a street. To produce the model, such approaches will take audio signals from various streets, analyze these signals' stochastic properties, and extract the most representative (and independent) features. However, these approaches don't separate the signals according to the scenes that make up the complex setting (cars, pedestrians, street bands, and so forth).

In reality, a street setting isn't a result of a stationary mix of different events but rather a complex mix of time-variant events. For example, the frequency and types of cars passing by change continuously. A recognition scheme can deal with this type of dynamic only if it can separate the street's stationary scenes from the transient scenes. Moreover, by modeling the scenes independently and establishing a relationship between them, we can define a higher-level context declaratively.

We model complex settings as an aggregation of distinct atomic scenes. To support declarative context aggregation, we provide a conceptual architecture that enables a systematic modeling and gradual reasoning of complex settings. Applying our architecture to auditory-based context recognition, we've modeled seven everyday situations with more than 20 atomic scenes, achieving high recognition rates for both the atomic scenes and complex settings.

A conceptual architecture for context recognition

Humans recognize complex settings by perceiving individual settings and examining the relationships between them. Their certainty of the perceived setting depends on how well they have gathered and interpreted data from their surroundings. It also depends on the presence or absence of some vital scenes that constitute the setting. A complex setting consists of individual settings unfolding in a certain order. Moreover, by combining individual settings from their memory, humans can imagine set-

1541-1672/08/$25.00 © 2008 IEEE
Published by the IEEE Computer Society

tings they have never experienced. For example, a person who never watched a symphony orchestra playing Beethoven can imagine it by combining pictures of individual scenes of an orchestra from his or her experience.

Our aim is to imitate human-like reasoning. Proper imitation will lead us to

- improve context recognition accuracy and
- declaratively define an entirely new setting by aggregating known individual scenes.

To this end, we propose the four-layered conceptual architecture in Figure 1.

The architecture's bottom layer (the raw-sensor-data layer) consists of an array of physical sensors embedded in mobile devices or carefully placed in physical environments.

The second layer extracts primitive features (contexts). A primitive context represents a single, indivisible aspect of a certain phenomenon or physical entity (device, place, person, and so forth). It's a meaningful interpretation of raw sensed data. Because it's primitive, it's extracted either from a single sensor or from multiple sensors representing the same aspect. Unlike a higher-level context, whose meaning is application-specific, a primitive context can be useful for recognizing several higher-level contexts.

The third layer constructs atomic scenes. The premise for this layer is that most everyday settings consist of distinct scenes, and multiple settings can have several scenes in common. If the system recognizes these scenes and stores their models separately in a knowledge base, it can reuse them to declaratively define complex settings for which it hasn't previously been trained.

For example, we can describe a meeting setting by the flipping of papers, conversations, and occasional whispers. We can describe a lecture by a monotonous oration, flipping of papers, occasional coughs, sporadic whispers, the sound of writing with chalk on a blackboard, and so forth. These two settings share the flipping of papers and whispering. A context-recognition system can therefore exploit this knowledge to accommodate the definition of a meeting or a lecture even though it has never been trained to recognize either of these two contexts.

The fourth layer handles context recognition. It employs a deterministic or probabilistic reasoning scheme or a combination of both (for more on these approaches, see the "Related Work in Context Recognition" sidebar on p. 60). It aggregates evidence from the third layer, establishes logical or probabilistic relations between the atomic scenes the system has already recognized, and computes a higher-level context. The layer takes into account domain knowledge of the mutual occurrence of the atomic scenes.

Common to all layers except the raw-sensor-data layer is the knowledge base. It comprises facts that constitute an application domain's vocabulary and a list of assertions about individual named entities in terms of this vocabulary. The vocabulary consists of concepts, which denote sets of entities, and relations, which denote binary relationships between these entities. The knowledge base also allows the building of complex descriptions of concepts and relations. The system uses this knowledge to extract meaningful features from sensors, classify atomic scenes, and model relationships between the atomic scenes to recognize higher-level contexts.

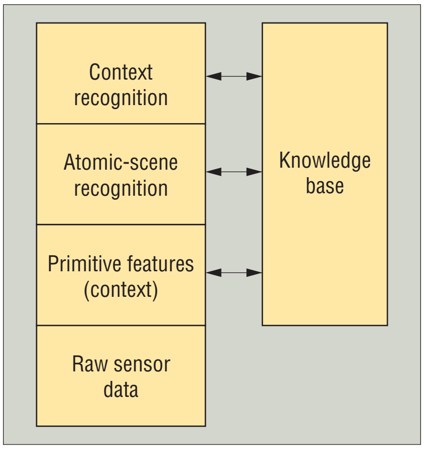

Figure 1. A conceptual architecture for recognition of complex settings. This architecture allows a recognition system to mimic human reasoning.

Auditory-based context recognition

We chose auditory signals for three reasons. First, among the human senses, hearing is second only to vision in recognizing social and conceptual settings; this is due partly to the richness in information of audio signals. Second, you can embed cheap but practical microphones in almost all types of places or mobile devices, including PDAs and mobile phones. Finally, auditory-based context recognition consumes significantly fewer computing resources than camera-based context recognition.

To better explain the implementation of our architecture for auditory-based context recognition, we offer here a summary of digital audio-signal processing.

Even though auditory-based context recognition is similar to speech recognition, there are several differences. For example, in speech recognition, knowledge of human perception (tone, pitch, loudness, and so forth) is useful to disambiguate an uttered speech. This is possible because

- the speaker isn't far from the microphone and speaks sufficiently loud, and
- no significant hindrance exists between the speaker and the microphone.

This isn't the case with auditory-based context recognition. First, the audio-signal amplitude representing a user's surrounding isn't appreciably large because the audio sources might be farther from the user (the microphone). Moreover, the device with the embedded microphone might be hidden in a suitcase or pocket. So, auditory-based context recognition can't achieve the same accuracy as speech recognition.

Extracting audio features

The statistical properties of audio signals representing most everyday settings aren't stationary. To extract features that represent temporal and spectral aspects, audio-based recognition systems divide the audio data stream into small time frames that can then be considered quasi-stationary. Some overlap between the frames is desirable; typically, the overlap is between 25 and 50 percent. A frame's duration is usually between 10 and 50 milliseconds, depending on the desired recognition accuracy and computation time. Further processing isn't necessary to extract temporal features, but at least two additional steps are necessary to extract spectral properties.

Related Work in Context Recognition

A context recognition (reasoning) process can be deterministic, probabilistic, or both. Deterministic context reasoning classifies sensed data into distinct states and produces a distinct output that can't be uncertain or disputable. Probabilistic reasoning, on the other hand, considers sensed data to be uncertain input and thus outputs multiple contextual states with associated degrees of truthfulness.

Several researchers have proposed probabilistic-reasoning techniques for context reasoning. These techniques differ according to the type of context they recognize and the types of sensors they employ.

Nicolas Moeënne-Loccoz, François Brémond, and Monique Thonnat proposed Bayesian networks to recognize various human activities on a street (aggressive behavior, casual talk, and play); they obtained sensed data from a camera.[1] Huadong Wu employed a camera and several microphones to reason about the attention of people during a meeting session.[2] He applied the Dempster-Shafer theory of evidence to combine data from microphones with data from an omnidirectional camera.

Jani Mäntyjärvi, Johan Himberg, and Pertti Huuskonen proposed *k*-means clustering and minimum-variance segmentation algorithms to process data from a skin conductance sensor, a microphone, a light sensor, an accelerometer, and a temperature sensor, to recognize a mobile device's status and its user's activity.[3] Device status refers to whether the device is in the user's hands, on a table, or inside a suitcase; user activity refers to walking, running, or going up or down a staircase.

Some researchers have focused particularly on processing audio signals to recognize various everyday human situations. Vesa Peltonen and his colleagues classified auditory scenes into predefined classes by employing two classification schemes: a 1-NN (1-nearest neighbor) classifier and Mel-frequency cepstral coefficients (MFCCs) with Gaussian mixture models.[4] The auditory scenes comprised several everyday outdoor and indoor situations (streets, restaurants, offices, homes, cars, and so forth). The features extracted from audio signals for classification were time and frequency domain features and linear prediction coefficients. Altogether, the classification systems classified 17 indoor and outdoor scenes with an accuracy of 68.4 percent.

For their experiment, Peltonen and his colleagues considered various configurations: a binaural setup (a Brüel & Kjaer 4128 head and torso simulator), a stereo setup (AKG C460B microphones), and a B-format setup, which contains 3D information of the audio event being recorded (SoundField MkV microphone). They recorded the sounds on a digital multitask recorder with a 16-bit, 48-kHz sampling rate and on a Sony (TCD-D10) digital audio tape recorder with a 16-bit, 48-kHz sampling rate.

Antti Eronen replaced the two classifiers that Peltonen and his colleagues used with hidden Markov models (HMMs) to imitate human hearing sensitivity and to increase recognition accuracy up to 88 percent.[5]

Ling Ma, Dan Smith, and Ben Milner also employed HMMs and MFCCs to recognize 10 auditory scenes.[6] By varying the hidden states of the Markov models, they achieved different recognition rates. With only three hidden states, the classifier achieved 78 percent context recognition; with 15 hidden states, it achieved 91.5 percent recognition. Remarkably, context recognition declined for more than 15 hidden states. Dan Smith, Ling Ma, and Nick Ryan extended this research by introducing a belief revision mechanism that increased the recognition rate to 92.27 percent and the number of recognized contexts to 12.[7]

Panu Korpipää and his colleagues employed a naive Bayesian classifier and an extensive set of audio features derived partly from the algorithms of the MPEG-7 standard.[8] They based the classification mainly on audio features measured in a home scenario. To collect the data, Korpipää and his colleagues used an extra-small sensor box attached to a shoulder strap of a backpack containing a laptop. When collecting scenario data, researchers wore the backpack. A cordless mouse controlled the measurement system to mark the scenario phases. The microphone was a small, omnidirectional AKG C 417/B.

With a resolution of 1 second in segments of 5–30 seconds and using leave-one-out cross-validation, Korpipää and his colleagues achieved a recognition rate of 87 percent of true positives and 95 percent of true negatives, averaged over nine 8-minute scenarios containing 17 segments of different lengths and nine different contexts. The reference accuracies measured by testing with training data were 88 percent (true positive) and 95 percent (true negative), suggesting that the model can cover the variability introduced in the data on purpose. Reference recognition accuracy in con-

Owing to the abrupt separation of neighboring frames, high-frequency components will emerge at both edges of each frame. This *frequency leakage* should be removed (or at least its effect should be minimized) through a *windowing* operation, a filtering process that multiplies each frame with a window function that decays rapidly toward the edges. Before this process, however, we want to smooth the spectrum and enhance the high-frequency components by passing the frames through a first-order, finite-impulse-response preemphasis high-pass filter:

$$s_{sp}(n) = s(n) - \mu s(n-1)$$

In this equation, $s_{sp}(n)$ is the improved nth sample of a frame, $s(n)$ is the original nth sample, $s(n-1)$ is the original $n-1$th sample, and μ is a unitless quantity, which normally ranges between 0.90 and 0.98. For the windowing operation, we use a standard Hemming window, which we can describe as

$$s_w(n) = \{0.54 - 0.46 \times \cos(2\pi(n-1)/(N-1))\} \times s_{sp}(n)$$

where $s_w(n)$ refers to the nth sample of a frame that has passed through a Hemming window and N is the number of samples in a frame.

Mel-frequency cepstral coefficients (MFCCs) are the most frequently used features for classifying auditory data. They represent frequency bands that are Mel-scaled to approximate the human auditory system's response more accurately than linearly spaced frequency bands obtained directly from a fast Fourier transform trolled conditions was 96 percent (true positive) and 100 percent (true negative).

Filip Bonnevier employed Bayesian networks to recognize 25 different contexts from 21 MPEG-7 features with a 69 percent recognition rate.[9] Interestingly, the context recognition ran on a pocket PC.

Table A summarizes the audio-based context-recognition schemes, their recognized contexts, and their recognition accuracies.

References

1. N. Moeënne-Loccoz, F. Brémond, and M. Thonnat, "Recurrent Bayesian Network for the Recognition of Human Behaviours from Video," *Proc. 3rd Int'l Conf. Computer Vision Systems* (ICVS 03), Springer, 2003, pp. 66–77.
2. H. Wu, "Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory," PhD thesis, Dept. of Computer Science, Carnegie Mellon Univ., 2003.
3. J. Mäntyjärvi, J. Himberg, and P. Huuskonen, "Collaborative Context Recognition for Handheld Devices," *Proc. 1st IEEE Int'l Conf. Pervasive Computing and Communications* (PerCom 03), IEEE Press, 2003, pp. 161–168.
4. V. Peltonen et al., "Computational Auditory Scene Recognition," *Proc. Int'l Conf. Acoustic Speech and Signal Processing* (ICASSP 02), IEEE Press, 2002, pp. 1941–1944.
5. A. Eronen, "Automatic Musical Instrument Recognition," master's thesis, Dept. of Information Technology, Tampere Univ. of Technology, 2001.
6. L. Ma, D. Smith, and B. Milner, "Context-Awareness Using Environmental Noise Classification," *Proc. 8th European Conf. Speech Communication and Technology* (Eurospeech 03), 2003, pp. 2237–2240.
7. D. Smith, L. Ma, and N. Ryan, "Acoustic Environment as an Indicator of Social and Physical Context," *Personal Ubiquitous Computing*, vol. 10, no. 4, 2006, pp. 241–254.
8. P. Korpipää et al., "Managing Context Information in Mobile Devices," *IEEE Pervasive Computing*, vol. 2, no. 3, 2003, pp. 42–51.
9. F. Bonnevier, "Audio Based Context-Awareness on a Pocket PC," master's thesis, Dept. of Electrical Eng., Stockholm Inst. of Technology, 2006.

Table A. Audio-based context-recognition schemes.

Authors	Primitive features (contexts)	Classifier	Recognition accuracy (%)	Context
Peltonen et al.[4]	Temporal, spectral, Mel-frequency cepstral coefficient (MFCC)	*k*-nearest neighbor (*k*-NN) and Gaussian mixture model (GMM)	68.4 (17 of 26 contexts)	Bathroom, street, church, car, supermarket, office
Eronen et al.[5]	MFCC	Hidden Markov model (HMM)	Between 61 and 85 (18 contexts)	Library, office, lecture, train, bus
Ma, Smith, and Milner[6]	MFCC	HMM	91.5 (10 contexts)	Bar, beach, bus, lecture, office, street, launderette
Smith, Ma, and Ryan[7]	MFCC	HMM	92.27 (12 contexts)	Bus, car, presentation, supermarket, train, office
Korpipää et al.[8]	MPEG-7	Naïve Bayesian classifier	88 (9 contexts)	Running, walking, music, speech, elevator, tap water, car
Bonnevier[9]	Spectral, temporal, MPEG-7	Bayesian network	69 (25 contexts)	Street, car, bus, cooking, TV, kitchen, living room

(FFT) or a discrete cosine transformation (DCT). Such representations allow context-recognition schemes to "perceive" their surroundings as humans would perceive theirs.

To obtain MFCCs, we perform an FFT; the result passes through a bank of triangular filters called Mel-filters (see Figure 2 on p. 62) to produce the Mel-spectrum. The number of filters can vary, but as a rule, speech recognition uses 23 filters. These filters are equidistant in the Mel-frequency domain, with a 50 percent overlap between adjacent filters. The following equation computes the center of each triangular filter:

$$c(x) = 2{,}595\left(x \cdot \log_{10}\left(1 + \frac{x}{700}\right)\right)$$

where $c(x)$ is the center of a triangular filter (in the frequency domain) that has taken the amplitude spectrum, x, of the FFT as its input.

Finally, we perform an inverse DCT on the natural logarithm of the filters' output to obtain the MFCCs:

$$c_i = \sum_{j=1}^{N} \ln\left|f_i\right| \cos\left(\frac{\pi i}{N}\left(j - \frac{1}{2}\right)\right),$$
$$0 \le i \le M$$

where c_i is the ith cepstral coefficient, f_i is the ith frequency component, N is the number of the triangular filters, and M is

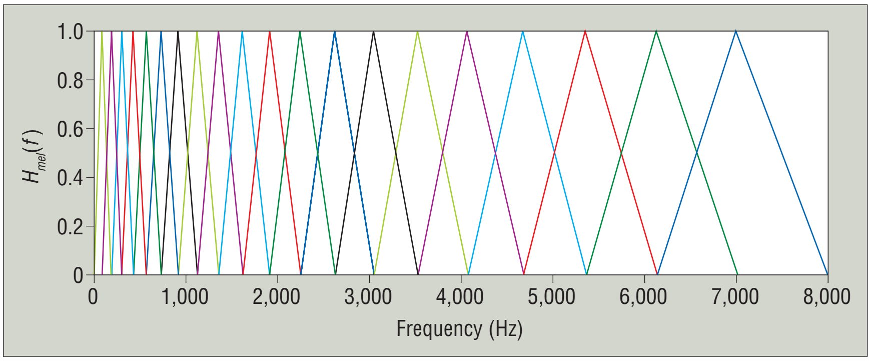

Figure 2. A triangular filter bank. The filters' response has a linear frequency spacing below 1,000 Hz and a logarithmic spacing above 1,000 Hz. $H_{Mel}(f)$ denotes a normalized magnitude spectrum.

Table 1. Higher-level contexts defined declaratively as aggregations of atomic scenes.

Higher-level context	Individual scenes
Office	Clacking of keyboard Conversation Mouse clicking Telephone conversation Telephone ringing
Cafeteria	Background noise Chair movement Clacking of cash register keys Clattering dishes Conversation
Library	Chair movement Clacking of keyboard Coughing Door opening and closing Flipping pages Mouse clicking Whispering
Tram	Station announcement signal Background noise Door-closing warning People getting on and off
Street	Moving cars People walking and talking Background noise
Lecture	Background noise Chair movement Coughing Flipping pages Oration Whispering Writing on a chalkboard
Train	Background noise People getting on and off Conversation

the number of the extracted MFCCs. In this way, we set the dimension of the feature vectors; the typical dimension is 13.

To weaken the effect of very low and high orders of the cepstral coefficients, we need to subject the MFCCs to a "band-pass filtering" process called *liftering*. The following equation displays a typical liftering function:

$$c_i = \left(1 + \frac{L}{2}\sin\left(\frac{\pi_i}{L}\right)\right)c$$

where c_i is the corrected cepstral coefficient, c is the ith uncorrected cepstral coefficient, and L is a liftering factor.

Recognition

Feature extraction quantizes the audio signal and transforms it into various characteristic features. This results in an n-dimensional feature vector representing each audio frame. A classifier then takes this feature vector and determines what it represents—that is, it determines an auditory scene.

Several recognition techniques are readily available, most of which we mention in the sidebar. The three most common are k-nearest neighbor (k-NN) classifiers, hidden Markov models (HMMs), and Bayesian networks.

Implementation

We selected these seven higher-level contexts: office, cafeteria, library, tram, street, lecture, and train. Table 1 lists these settings along with the associated atomic scenes.

We chose the atomic scenes on the basis of how well they represented the higher-level settings and how accurately they could be recognized.

The raw-sensor-data layer consisted of commonplace microphones embedded in ordinary laptop PCs during the training and test phases. Moreover, we recorded the audio signals without much preparation to imitate how users handle their mobile devices while moving or carrying out other more important activities.

We implemented the second layer by adopting the OC-volume framework (http://ocvolume.sourceforge.net). Even though the framework was initially intended for speech recognition, we could reuse it for extracting MFCCs and for vector quantization, using the LBG (Linde, Buzo, and Gray) algorithm.[1] However, we had to modify the algorithm to

- model time dependency in the audio signals and
- increase the signals' bandwidth to accommodate the surrounding noise's dominant frequencies.

As a result, we could consider ranges of frequencies between 30 and

Table 2. Recognition accuracy of atomic scenes.

Atomic scene	Recognition rate (%)	Deviation (%), with incorrect classifications
Car	100	0
Flipping pages	100	0
Door opening and closing	91	Background noise, Tram: 9
Chair movement	80	Background noise, Tram: 20
Door-closing warning	100	0
Clattering dishes	50	Coughing: 50
Coughing	100	0
Background noise, Tram	87.5	Oration: 12.5
Writing on a chalkboard	80	Whispering: 20
Whispering	56	Background noise, Lecture: 44
Oration	75	Background noise, Cafeteria: 10 Conversation, Office: 5 Background noise, Lecture: 5
Conversation	72	Oration: 18 Background noise, Train: 10
Background noise, Street	58	Writing on a chalkboard: 23 Background noise, Cafeteria: 19
Background noise, Lecture	40	Whispering: 60
Mouse clicking	94	Flipping pages: 4 Clacking of keyboard: 2
Background noise, Train	100	0
Station announcement signal	0	Poorly captured audio signal
Clacking of keyboard	79	Mouse clicking: 10 Background noise, Library: 6 Door opening and closing: 3 Coughing: 1 Background noise, Train: 1
Conversation, Telephone	100	0
Background noise, Cafeteria	65	Conversation: 25 Station announcement signal: 5 Chair movement: 5
Background noise, Library	62	Clacking of keyboard: 32 Mouse clicking: 6
Overall recognition rate	**69.92**	

10,000 Hz. For speech recognition, the frequency of interest is below 3,400 Hz.

To realize the atomic-scene layer, we chose a *k*-NN classifier because of its simplicity. It also classifies a large number of scenes in an acceptable recognition time. The classifier performs a class vote among the *k*-nearest neighbors on a point to be classified. A Euclidean distance, *d*, between the points determines which atomic scenes are represented by the extracted MFCCs. We set $k = 1$. Vesa Peltonen and his colleagues demonstrated that classification with *k* greater than 1 yields no significant improvement in recognition accuracy.[2]

Table 2 displays the atomic scenes we could recognize, the recognition accuracy, and the deviation, with the atomic scenes that were wrongly recognized. The quality of the recognized atomic scenes depended on how distinct they were from other atomic scenes. It also depended on the recorded audio signal's quality.

We chose a Bayesian network to model relationships between the higher-level contexts and the atomic scenes and to recognize a higher-level context. We used the JavaBayes framework (www.cs.cmu.edu/~javabayes/index.html) to implement the knowledge base and the context-recognition layer. The knowledge base stores models of the Bayesian network structure as well as conditional-probability distributions.

The Bayesian classifier establishes a network based on the atomic scenes recognized in the lower layer. We applied heuristic

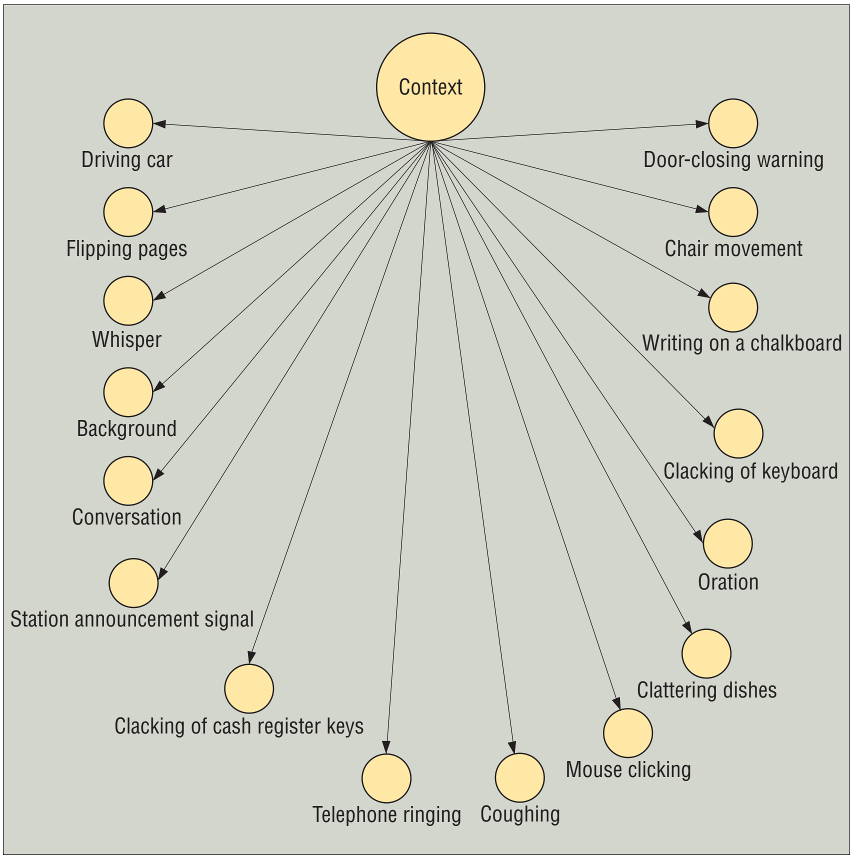

Figure 3. A Bayesian network for establishing conditional dependencies between higher-level settings and atomic scenes. The child nodes represent the atomic scenes, and the parent node represents the higher-level contexts. Altogether, the parent node can have seven different values depending on the network configuration.

observations for establishing the conditional dependencies between the atomic scenes and the higher-level contexts. Figure 3 shows our Bayesian network.

Bayesian networks apply Bayes's theorem to model probabilistic relationships among distinctions of interest in uncertain reasoning. The networks are directed acyclic graphs (DAGs) in which nodes represent random variables and a directed arrow represents a conditional dependency between the variables. A particular configuration of a Bayesian network refers to an instantiation of the random variables with values from a 2D value vector. A particular configuration's likelihood is determined by the sum of the products of the associated conditional probabilities.

A Bayesian network obeys the Markov condition for mathematical and computational tractability. So, a node is conditionally independent of its nondescendants given its parent in *G*, the network's graph topology. Mathematically, we express this as

$$p\left(n_1, n_2, \ldots, n_j\right) = \prod_{j=1}^{N} p\left(n_j \mid parent\left(n_j\right)\right)$$

where n_1, n_2, …, n_j are the possible values of the network's random variables, and *p* refers to probability.

Once we establish a Bayesian network and define the degree of independence between random variables, even partially, it's possible to carry out three essential tasks.[3] First, because the model encodes dependencies among all variables, it can readily reason about situations where some data entries are missing. Second, we can train the network to learn causal relationships and hence use it to understand a problem domain and to predict the consequences of intervention. Finally, because the model has both causal and probabilistic semantics, it's ideal for combining prior knowledge (which often comes in causal form) and data.

Discussion

Table 3 lists our higher-level contexts and the corresponding atomic scenes that contribute to their recognition. The last column lists the normalized percentage of each atomic scene's contribution. The percentage doesn't add up to 100 percent because the list doesn't include erroneous atomic scenes.

The higher-level context with the lowest recognition rate is a street (37 percent). In fact, the spurious cafeteria context had higher recognition accuracy—47 percent. Interestingly, the Bayesian classifier could recognize a cafeteria with 100 percent accuracy without mistaking it for a street or another contending setting. This implies that context recognition is asymmetric—a context's recognition accuracy depends on not only how well it's represented by the atomic scenes but also whether the captured auditory test signal typically represents the setting. For our case, for example, the test signal came twice from a street with little activity, and the activities at a nearby cafeteria dominated the recording. Predictably, this led to a wrong conclusion.

On the other hand, page flipping might seem difficult to recognize because it isn't loud. We could, however, recognize it with 100 percent accuracy. This is because the atomic scene was associated with a lecture and a library, where the background noise and other atomic scenes could be distinctly discerned. Moreover, as we trained and tested our system, we placed a laptop with a microphone near the user who was reading and flipping pages.

The least-recognized scene—in fact, the system didn't recognize it at all—was chair movement in a library. The system sometimes mistook an oration in a lecture room for a conversation in an office or cafeteria, which is understandable.

We were interested in comparing our results with others', but this wasn't easy. Some research reports conceal a wealth of information. Maybe this is because recognition accuracy depends on not only the particular schemes or features employed but also many other factors. To begin with, it depends on the types of contexts to be recognized. The larger and more similar the context types, the harder it is to distinguish between them. Recognition accuracy also depends on the test signal's length, the audio signal's sampling rate, the

Table 3. Recognition of complex settings by aggregating atomic scenes.

Higher-level context	Correctly and incorrectly recognized contexts (%)	Distribution of the correctly recognized atomic scenes (%)
Library	Library: 67 Office: 33	Clacking of keyboard: 25.17 Mouse clicking: 23.87 Door opening and closing: 20.86 Chair movement: 7
Office	Office: 90 Train: 10	Conversation, Telephone: 70.7 Clacking of keyboard: 25.17 Mouse clicking: 3.7
Cafeteria	Cafeteria: 100	Background noise, Cafeteria: 51.86 Conversation: 32.8 Clattering dishes: 7.4
Street	Street: 37 Cafeteria: 47 Lecture: 10 Train: 5	Moving cars: 51.49 Chair movement: 10.89 Background noise, Street: 10 Conversation: 8.49 Background noise, Cafeteria: 7.4
Tram	Tram: 75 Cafeteria: 25	Background noise, Tram: 70 Station announcement signal: 7
Train	Train: 40 Library: 40 Lecture: 15 Office: 5	Background noise, Train: 87.67 Conversation: 9.4
Lecture	Lecture: 100	Oration: 36.53 Chair movement, Lecture: 26.03 Background noise, Lecture: 15.19 Writing on chalkboard: 8.22 Whispering: 6.4
Overall recognition rate	**72.71**	

MFCCs' size, and the size of the code book of the vector quantization process. Subsequently, a trade-off always exists between recognition time and recognition accuracy.

More important, the recording devices used and the audio signal's length and duration influence context-recognition accuracy. Using expensive, bulky, and power-hungry audio devices might yield remarkable accuracy, but using them in everyday situations, particularly in mobile environments, isn't feasible.

The research that comes closest to ours is that of Peltonen and his colleagues and Antti Eronen (see the sidebar). Our atomic-scene-recognition accuracy is similar to theirs, but we achieved recognition accuracy through commonplace microphones and ordinary laptop computers as compared to the sophisticated devices they used to record audio signals. Moreover, our approach can be generalized to accommodate sensors other than microphones, while their approaches are limited to audio-based context recognition.

Our experience demonstrates the difficulty of context recognition using a single context source—namely, an audio signal. Humans aptly apply other faculties besides hearing to appropriately perceive their surroundings. This justifies the need for heterogeneous sensing.

We're interested in investigating the possibility of deploying—at least in part—audio-signal-processing algorithms on wireless sensor nodes. This will enable us to gather and process surrounding acoustic information and to better interface the physical world with the virtual world. ■

The Authors

Waltenegus Dargie is a researcher at the Technical University of Dresden. His research interests include ubiquitous computing, context-aware computing, wireless networks, and digital signal processing. Dargie received his PhD in computer engineering from the Technical University of Dresden. Contact him at waltenegus.dargie@tu-dresden.de.

Tobias Tersch is a software developer at the Sidon Software and Engineering Service-Providing Company. His research interests include context awareness and smart systems. Tersch received his diploma in computer science from the Technical University of Dresden. Contact him at tobias.tersch@web.de.

References

1. H.-Y. Chang et al., "Performance Improvement of Vector Quantization by Using Threshold," *Advances in Multimedia Information Processing—PCM 2004*, LNCS 3333, Springer, 2005, pp. 647–654.
2. V. Peltonen et al., "Computational Auditory Scene Recognition," *Proc. Int'l Conf. Acoustic Speech and Signal Processing*, 2002.
3. D. Heckerman, "A Tutorial on Learning with Bayesian Networks," *Learning in Graphical Models*, M.I. Jordan, ed., MIT Press, 1999, pp. 301–354.