Title: A Toolbox for Animal Call Recognition

Authors: Michael Towsey, Birgit Planitz, Alfredo Nantes, Jason Wimmer and Paul Roe

Affiliation: Microsoft QUT eResearch Centre, Queensland University of Technology, Brisbane, Australia

Abstract:

Monitoring the natural environment is becoming increasingly important as habit degradation and climate change reduce the world's biodiversity. We have developed tools, applications and processes to assist ecologists with the collection and analysis of acoustic data at large spatial and temporal scales. One of our key objectives is automated animal call recognition, and our approach has three novel attributes. First, we work with raw environmental audio, contaminated by noise and artefacts, and containing calls that vary greatly in volume depending the animal's proximity to the microphone. Second, initial experimentation suggested that no single recogniser could deal with the enormous variety of calls. Therefore, we developed a toolbox of generic recognisers to extract invariant features for each call type. Third, many species are cryptic and offer little data with which to train a recogniser. Many popular machine learning methods (e.g. Hidden Markov Models, Neural Networks and Support Vector Machines) require large volumes of training and validation data, and considerable time and expertise to prepare. Consequently we adopt bootstrap techniques that can be initiated with little data and refined subsequently. In this paper, we describe the recognition algorithms in our toolbox and present results for their performance on real ecological problems.

1 Introduction

The increased availability, power and storage capacity of computing hardware have made it feasible to gather large volumes of audio data for ecological analysis. In addition, enhanced web services have made it possible to bring that audio data directly into the laboratory rather than have ecologists go into the field. However, it is impossible for ecologists to listen to even a small fraction of the audio data made so easily available (Agranat 2009). Some degree of automated assistance is essential. In conjunction with ecologists, our laboratory has developed an online service <http://sensor.mquter.qut.edu.au> that offers a variety of tools for the analysis of environmental recordings. In a previous report, we described various aspects of our sensor network for collecting audio data (Lau et al. 2008). In this report we describe our approach to the problem of automated analysis and, in particular, to automated animal call recognition.

Perhaps due to the importance of birds as indicator species of environmental health, there is already a considerable body of work published on the detection of bird vocalisations (Acevedo et al. 2009; Anderson et al. 1996; Brandes 2008; Cai et al. 2007; Chen & Maher 2006; Juang & Chen 2007; Kwan et al. 2004; McIlraith & Card 1997; Somervuo et al. 2006). A common approach has been to adopt the well-developed tools of Automated Speech Recognition (ASR), which extract Mel-Frequency Cepstral Coefficients (MFCCs) as features and use Hidden Markov Models (HMMs) to model the vocalisations. Unfortunately it is not so easy to translate ASR to the analysis of environmental recordings because there are far fewer constraints in the latter task. Two issues are noise and variability. ASR tasks are typically restricted to environments where noise is tightly constrained, e.g. over the telephone. By contrast, environmental acoustics contain a wide variety of non-biological noises having a great range of intensities and a variety of animal sounds

that have nothing to do with the task at hand. Furthermore, the sources can be located any distance from the microphone. Secondly, despite its difficulty, ASR applied to the English language requires the recognition of about 50 phonemes. By contrast, bird calls offer endless variety; variety of call structure between species, variety between populations of the one species and variety within and between individuals of the one population. Many species have multiple calls (in this paper we do not distinguish bird calls form bird songs) and many are mimics. To give some indication of the difficulty of bird call recognition, a state-of-the-art commercial system using an ASR approach that has been under development for more than a decade, achieves, on unseen test vocalisations of 54 species, an average accuracy (defined as the average of precision and recall) of 65% to 75% (Agranat 2009). This accuracy may not be sufficient for some applications. Furthermore, the software requires the user to tune many parameters which require a good understanding of the underlying algorithms. Some work has been done in the urban setting on the recognition of acoustic events in auditory scene analysis but these tasks suffer from exactly the same difficulties (Cowling & Sitte 2003; Temko et al. 2006; Zhuang et al. 2008).

Our approach to animal call recognition has three attributes which help to distinguish it from many previous approaches to the same problem: real world data, multiple recognition methods and limited training data.

1. Real world data

The results we describe are obtained with real world data that has not been cleaned of artefacts. There is a world of difference between reporting classification results on carefully cleaned data with balanced training and test sets versus the raw recordings to which ecologists actually listen. Constructing a data set containing equal numbers of each call type that have been manually cut from recordings to exclude extraneous

noise is not a realistic approximation to the real situation of very uneven class numbers and low call density in arbitrary background noise.

2. Multiple recognition methods

After an initial period where we adapted ASR methods to animal call detection, we came to the conclusion that a one-algorithm-fits-all approach could not deal with the enormous variety of environmental acoustic events. To address this issue, we developed a toolbox of generic recognisers that identify invariant features in calls of interest. For bird calls consisting of a single syllable (e.g. the currawong, see Figure 1(a)), extracting MFCCs and training HMMs may still be a suitable choice. However the 'tool-box' approach allows us to mix and match feature extraction with classifiers to suit generic call types.

3. Limited training data

Despite the demonstrated accuracy of machine learning methods such as Neural Networks (NN) and Hidden Markov Models (HMM) on standard datasets, these methods do not necessarily adapt well to the real world of environmental recordings. Many bird species are cryptic and the large amounts of data required to train a NN or HMM are not available. It is more practical to adopt methods that require just one or a few instances of a call type. This approach is effective for species whose calls vary little within and between populations (e.g. the Lewin's Rail (Lau et al. 2008)). Another consideration is the practicality of training of multi-class classifiers. Training a 50 bird-call classifier for a given locality may be possible but it is not practical if it must be repeated every time the ecologist wishes to incorporate a new call instance or call class.

In this paper we describe a 'toolbox' of animal call recognition techniques developed in our laboratory. (It is available at <http://sensor.mquter.qut.edu.au/>.) The main motivation for our work has been to provide tools that are easy to use (compared to the more sophisticated techniques of ASR) and which are well adapted to the real problems confronted by ecologists wishing to process many hours of environmental recordings. This is not to say that the more sophisticated tools have no value. Rather our tools could be viewed as filters to highlight points of interest in long recordings.

We limit ourselves to terrestrial animals – more specifically we exclude marine animals whose calls present a different set of problems (Rickwood & Taylor 2008). In addition we avoid birds that mimic. In theory, any animal call used for communication purposes should contain invariant features but in practice the recognition of mimics can be a very difficult task, even for humans.

In Section 2, we describe some different call structures, their invariant features and the recognition algorithms appropriate for them. In Section 3 we describe experiments with datasets obtained for selected animal calls and present our results. We conclude the paper with a discussion of our ongoing work in expanding the toolbox.

2 Call Detection Algorithms

2.1 Call Structures

Many animal calls have a hierarchical structure. A complex bird call, for example, may be divided into phrases, the phrases into syllables and the syllables into one or more elements (Catchpole & Slater 1995). Each element may take the form of a whistle (single tone), chirp (slowly modulated tone), whip (rapidly modulated tone), click (vertical line in spectrogram), vibrato, shriek, stacked harmonics (simultaneous

multiple tones) or buzz (rapidly repeated click) (Catchpole & Slater 1995). The same syllable can be repeated multiple times. Figure 1 illustrates the structure of calls used in this work. Each image has been extracted from a spectrogram - the x-axis represents time, the y-axis frequency, and the grey scale represents acoustic intensity.

The currawong call (Figure 1(a)) consists of a frequency modulated whistle. The curlew call consists of multiple syllables having simple to complex structure, with or without harmonics. The example shown in (Figure 1(b) is a simple modulated tone. The male koala has a complexly structured call consisting of inhalations and exhalations lasting for 30 seconds or more. A small portion of its call (illustrating three oscillatory exhalations) is shown in Figure 1(c)). The cane toad and gecko have multi-syllable calls. The former consists of a click rapidly repeated for 20 seconds or more and the latter consists of some five to six clicks slowly repeated (Figure 1(d-e)). The ground parrot call consists of about 10-13 syllables, each one a descending chirp. Successive syllables increase in pitch (Figure 1(f)). The whistle and whip of a whip bird call is illustrated in Figure 1(g)). The whip may be either ascending or descending. Three examples of stacked harmonic calls are shown in Figure 1(h-j)). Human vowels typically display a dense stack of harmonics in a spectrogram.

While all calls exhibit some form of variability (e.g. the periodicity of the gecko call depends on temperature (Marcellini 1974), each call type nevertheless has one or more invariant features on which recognition can depend. The same can even be said of certain diffuse, apparently structureless events such as wind (Figure 1(k)) and canopy rain (Figure 1(l)).

Our toolbox contains a suite of classifiers each of which we have found appropriate for different call structures and syllable types (see list in Table I). Call structure dictates feature selection, which in turn dictates the classification algorithm. For example to detect calls (or parts of calls) consisting of a single-syllable frequency-modulated whistle, we use its STFT spectral image for template matching. Spectral templates are of limited use, however, if the call is variable and in these cases the invariant features of a call must be found. The whistle in a whipbird call, for example, varies in pitch but always consists of a horizontal 'track' found within a defined frequency band. The following section describes the variety of recognition algorithms in our toolbox in more detail.

2.2 Feature Extraction and Classification

Calls are recorded using the hardware described in Section 3.1. Recordings are sampled at 22,050 Hz (or subsequently down-sampled to this value) and a bit rate of 16. We outline the steps used by each recognition algorithm below.

2.2.1 MFCCs and Hidden Markov Models

As noted in the introduction we did not find the use of MFCC features and HMMs to work well for our recognition tasks. Yet we started with this approach because it has been reported in the literature for bird call recognition. We include it in this report simply to compare its performance with other methods.

We implemented the Hidden Markov Model Toolkit (HTK, <http://htk.eng.cam.ac.uk/>), a freely available software library for designing, training

and testing HMMs (Young et al. 2006). Although tailored for speech recognition tasks, HTK has also been applied to biological sequences (Akhtar et al. 2007) and bird call recognition (Trifa et al. 2008). HTK automatically extracts MFCC features before training the HMMs. We use most of the default parameter settings, in particular 50% frame-overlap and 12 cepstral coefficients. The frequency band is constrained to match the call to be recognized. Rather surprisingly, we obtain best results when omitting signal energy, and the dynamic delta and acceleration features.

This method represents calls as sequences of observations, each observation (a vector of cepstral coefficients derived from a frame) being treated as an emission from a dynamical system whose state transitions are described by a Markov process. In a simple Markov process each observation would be assigned to a unique system state but animal calls are too variable for such a restrictive model. In an HMM each observation is modelled as a probabilistic function of system state and each call type is modelled as a sequence of states. An HMM classifier returns the probability that the observed sequence would be emitted by a given call model.

Important parameters are the number of model states, the number of emission categories, the number of training iterations and the estimation of an HMM representing background noise. There are two possibilities for this last: 1) estimate a noise model from the silence periods in the training instances, or 2) estimate a noise model from separate recordings appropriate to the operational environment. We tried both approaches and had more success with the latter.

2.2.2 Spectrograms

All the remaining call recognition algorithms employ features derived from spectrograms prepared using the Short-Time Fourier Transform (STFT). The choice of parameters is guided by those typically used in ASR and by (Brandes 2008; Brandes et al. 2006). The signal is framed using a window of 512 samples (23.2ms) which offers a reasonable compromise between time and frequency resolution. A Hamming window function is applied to each frame prior to performing a Fast Fourier Transform (FFT), which yields amplitude values for 256 frequency bins, each spanning 43.07 Hz. The spectrum is smoothed with a moving average filter (window width = 3). The amplitude values (A) are converted to decibels (dB) using $dB = 20.\log10(A)$. Note that the dB values at this stage are with respect to a hypothetical signal having unit amplitude in each frequency bin. Frame overlap varies from 0 to 75% depending on the amount of fine structure in the call.

2.2.3 Binary Template Matching

Some calls, in particular frequency modulated whistles (Figure 1(a-b)), can be recognised using a simple binary template. The user marquees the call of interest in a noise reduced spectrogram and extracts a binary representation using an intensity threshold (typically around 4 dB above background). The template's *on*-cells define the call of interest and its *off*-cells contribute to an error function. Our template extraction tool allows manual editing to clean up background noise and to idealize the shape.

Although a binary template does not model the variations in acoustic intensity distributed through time and frequency, this lack of specificity can be an advantage if it generalises over irrelevant call variability.

The template is passed over all the frequency bins in a user specified band and a score is calculated for each temporal location (i.e. frame). The score may be calculated in a number of ways. To pick out faint calls from background noise (that is, to minimise false negatives), we use the difference in mean intensity between template *on*-cells and *off*-cells:

$$\text{Score} = \sum_{\text{on}} intensity/c_{\text{on}} - \sum_{\text{off}} intensity/c_{\text{off}},$$

where $\sum_{\text{on}} intensity$ and $\sum_{\text{off}} intensity$ are the sums of the acoustic intensity in the *on* and *off* cells respectively and c_{on} and c_{off} are the count of *on* and *off* cells respectively. 1-2 dB is a minimum perceptible difference and we find a suitable threshold score to adjust the recall/sensitivity trade-off lies in the 2.0 to 5.0 dB range.

Alternatively, we have used:

$$\text{Score} = (\sum_{\text{on}} intensity - \sum_{\text{off}} intensity)/c_{\text{on}},$$

which has the advantage that it imposes stricter penalties on the noise intensity in the *off*-cells (assuming $c_{\text{off}} > c_{\text{on}}$) and can be used to lessen false positive errors. Both metrics work on the principle of matching shape and intensity profiles in images (Brunelli 2009).

2.2.4 Oscillation Detection

Many animal calls consist of a repeating or oscillating single syllable, for example the Lewin's Rail and the gecko. Even the male koala bellow has a characteristic

oscillatory structure that can be used for identification purposes. Oscillation Detection (OD) is performed on the spectrogram using the Discrete Cosine Transform (DCT).

The DCT is a highly effective and sensitive technique, but should be used with appropriate caution. We apply it to the time series in each frequency bin of the spectrogram prior to noise removal since our noise removal algorithm tends to remove faint oscillations that are nevertheless detectable by DCT (Towsey et al. 2010). The same limitations apply to the DCT as to the FFT, in particular the requirement for signal stationarity over the duration of the DCT. In addition, brief acoustic impulses cause the DCT to return spurious oscillatory content.

Two significant parameters for the DCT are the time duration and threshold amplitude required to register a 'hit'. The optimum value for time duration will depend on the expected oscillation rate and stationarity of the call to be detected. The maximum DCT amplitude is obtained after normalisation of the DCT coefficients to unit length. Experimentation has shown that threshold values between 0.4 and 0.6 are suitable. In practice, it is not difficult to determine appropriate parameter values using training data as long as call variability (e.g. the oscillation rate) is within definable limits.

Call identification depends on recognising concentrations of oscillation 'hits' within the user-defined constraints of frequency band, time duration and oscillation rate. These constraints are essential to cull false positive detections and are effective if the sought call falls within characteristic bounds. Typical parameter values for male koala, gecko and cane toad are shown in Table II. The recall/sensitivity trade-off is controlled by adjusting the fraction of 'hit' bins (within the call's frequency band)

required for a positive detection. For a more detailed description of this method see (Towsey et al. 2010).

2.2.5 Segmentation

The speed of the OD algorithm can be significantly improved by avoiding periods of silence. Our segmentation algorithm extracts acoustic content by calculating the acoustic energy in each signal frame in the frequency band of interest. We determine a baseline (modal) frame-energy and its standard deviation using the method described in (Towsey et al. 2010). Segmentation involves retaining consecutive frames whose energy exceeds a user-threshold (defined in terms of noise standard deviations above the modal noise level). These are expected to be frames that contain potential calls of interest. In this work, we apply segmentation only in conjunction with OD, in which context a filter is required to smooth over oscillatory gaps. The width of the smoothing window is given by:

width = 1 / minimum expected oscillation rate for calls of interest

To further reduce computation, frame energy is calculated only for the middle quarter of the user-defined frequency band.

2.2.6 Acoustic Event Detection

We developed Acoustic Event Detection (AED) to extract information about the distribution of acoustic energy in a spectrogram using the minimum of prior knowledge. AED processes the spectrogram like an image and returns rectangular marquees around isolated acoustic events.

For a spectrogram:

Step 1: Apply a 2D smoothing Wiener filter.

Step 2: Noise reduce by removing the modal noise intensity calculated for each frequency band.

Step 3: Convert the noise-reduced spectrogram to a binary image with a user-defined intensity threshold. An effective threshold typically ranges from 6-9 dB, which corresponds to values in the literature (Brandes 2008). Note that a minimum perceptible acoustic difference is around 1-2 dB.

Step 4: Because a single threshold may fragment low intensity events, we join event pixels separated by N or fewer pixels in the vertical or horizontal directions. By default, N = 1.

Step 5: Place a rectangular marquee around the outer limit of each event, defined as any group of contiguous (in 8 directions) black pixels in a white background.

Step 6: The previous steps will incorrectly join acoustic events that overlap, but which, to both ear and eye, are due to separate sources. We are able to detect and separate most overlapping events by examining intensity distributions within anomalously large events.

Step 7: Cull small events - those whose area falls below a user defined threshold (a default value of 200 pixels was usually satisfactory for a typical spectrogram with 256 frequency bins and a frame rate of 86 per second).

Note that there are two important user defined parameters in this algorithm: the intensity threshold used to convert the spectrogram to binary form and the small area threshold, below which events are eliminated. Other thresholds were estimated from the data itself.

2.2.7 Event Pattern Recognition

Some multi-syllable animal calls can be modelled by the 2D distribution of their component syllables in the spectrogram rather than by the actual content of those syllables. We applied this technique, which we termed Event Pattern Recognition (EPR), to the recognition of ground parrot calls for which it proved effective even when the spectrogram was contaminated by numerous other 'noise' events. This would not be possible using MFCC features.

A single training example is sufficient to define a call in terms of its component events (see Figure 2). In recognition mode, this call template is passed over the acoustic events extracted from a test signal (hereafter referred to as the 'test' events) using the AED algorithm in Section 2.2.6. To limit unnecessary computation, only 'test' events were considered whose centroid lay within a user defined frequency band (3.5 - 4.5 kHz for ground parrots). For any match between template and test-events, the match-score is calculated as the average overlap between the template events and the closest 'test' events whose centroids fall within the bounds of the template. The fractional overlap between a single template event and its closest 'test' neighbour is given by:

$$\text{Overlap} = \tfrac{1}{2}(x/T + x/E)$$

where x = the overlapped area (in pixel units), T = the (pixel) area of the template event and E = the (pixel) area of the 'test' event. The overlap fraction lies between 0.0 (no overlap) and 1.0 (exact coincidence). The average overlap of all the events in the template gives rise to a score between 0.0 (complete mismatch of all events—actually not possible because at least the first template event must find some overlap)—and 1.0 (complete coincidence of 'test' and template events).

The recall/sensitivity trade-off can be adjusted using an overlap threshold in the range [0, 1]. The optimum value for this threshold should be derived from an ROC curve and strictly speaking the data required to obtain the ROC curve has the status of training data. In our case we had so few calls, even in a 6 hour recording, that the results described in Section 3.3.7 were obtained by optimizing the threshold on the available data.

2.2.8 Spectral Peak Tracking and Syntactic Pattern Recognition

Spectral Peak Tracking (SPT) isolates the traces of spectral ridges in a spectrogram (Chen & Maher 2006). It is most useful for the recognition of acoustic events defined by clean whistles. Fortunately many bird calls have this characteristic. SPT is not useful to detect parrot shrieks or diffuse events such as made by wind and rain.

As originally published, SPT was not well equipped to detect near-vertical tracks or whips such as made by several Australian birds (for example the whipbird, Figure 1(d) and the Golden Whistler). Here we describe a modification to the SPT algorithm for the better detection of whips.

Step 1: Smooth the input spectrogram using a 2D Wiener filter.
Step 2: Noise reduce the spectrogram. (Towsey & Planitz 2010).
Step 3: Identify (near-) horizontal tracks by identifying maxima in the spectral frame.
Step 4: Identify (near-) vertical tracks by identifying maxima in the time series for each frequency bin.
Step 5: Overlap the resulting horizontal and vertical tracks.

Step 6: Remove tracks (any arbitrarily branched set of 8-directionally consecutive pixels) whose total pixel count is less than a threshold (default value = 15 pixels for spectrograms prepared with default parameter values).

The two important parameters are the dB threshold for noise removal (step 2) and the short-track threshold (step 6). A difficulty with the algorithm is that it tends to highlight echoes (as wisps trailing a call) not otherwise obvious in the original spectrogram.

Syntactic Pattern Recognition (SPR) depends on the ability to represent a pattern as a sequence of symbols selected from a finite alphabet, each symbol representing a 'primitive' element of the composite pattern (Bunke & Sanfeliu 1990). This permits the representation of complex sequential patterns more accurately than can be achieved with 'flat' feature vectors of fixed dimensionality. The whipbird call has a simple two component structure (whistle followed by whip; see Figure 1(g)) that suggests the possibility for two primitives, a horizontal line segment and a near-vertical line segment. Recognition depends on detecting a sequence of horizontal primitives (the whistle) followed by vertical primitives (the whip) where the whistle can be of varying durations and frequency and the whip can be either ascending or descending. In our implementation, the notion of a whipbird grammar is implicit in the scoring algorithm.

Step 1: The input to SPR is a spectrogram in which spectral tracks have been highlighted using SPT as described above.

Step 2: Apply a user-defined intensity threshold to convert the spectrogram to a binary matrix.

Step 3: Identify the location of horizontal and vertical line primitives in the spectrogram. The identification of line primitives depends on an additional two parameters, the length of the line primitive and a sensitivity threshold. A primitive is found when the per cent of *on*-cells in a line segment exceeds a threshold.

Step 4: Assign a horizontal line-primitive score to each frame. We search within a user-defined bandwidth and time-period for horizontal primitives. The horizontal score for frame *N* is the fraction of frames over a *previous* (user-specified) time period traversed by a horizontal primitive.

Step 5: Assign a vertical line-primitive score to each frame. The vertical score for frame *N* is the fraction of cells in the *subsequent* rectangle (enclosed by the user-specified time period and frequency band of a whip) traversed by a vertical primitive.

Step 6: The whipbird score for frame *N* is the average of its horizontal and vertical scores. Since both scores lie in the interval [0, 1] a threshold in [0, 1] can be used to adjust the recall/sensitivity trade-off for the combined score.

Step 7: A hit is predicted where the score exceeds the user defined threshold (as described in step 6) for the number of consecutive frames set by the user for a whip duration (in step 5).

2.2.9 Harmonic Detection

Many animal calls display harmonics above a fundamental tone. Furthermore, the harmonic tracks often trace paths nearly parallel to the fundamental. Typical examples are the female koala (Figure 1(h)), human vowels (Figure 1(i)), crows (Figure 1(j)) and squeaky door hinges. A serious difficulty with using harmonics as a feature for

call recognition in environmental recordings is that higher harmonics drop out at a distance. Consequently the spectral pattern of a nearby bird is very different from that of the same bird at a distance. In ASR this problem does not arise because the speaker talks into a microphone. In environmental recordings, calls will be uttered at any distance. Nevertheless, Harmonic Detection (HD) can be useful if one knows the calls will be uttered close by or if one wishes to determine the proximity of a source by the number of high frequency harmonics.

Although the DCT is used in ASR to extract cepstral coefficients from spectra, we did not find this approach suitable for environmental recordings. For bird calls consisting of a pure whistle (one frequency), the whistle presents to the DCT like an impulse and therefore returns spurious 'high harmonic' content. Instead we counted the number of harmonic tracks appearing within a user defined bandwidth. The score for a given frame is a measure of the average intensity of the harmonic peaks (dB above background) in a noise reduced spectrum (Towsey & Planitz 2010) discounted by a function of the difference between the number of observed and expected harmonic peaks:

$$\text{Frame HD score} = (\textstyle\sum_n a_n / N).w,$$

where a_n is the amplitude of the n^{th} spectral peak, N is the number of observed spectral peaks in the user defined bandwidth, and w is a weighting factor that is a function of the difference between the observed number of peaks, N, and the expected number E:

$$w = 1.0 \quad \text{if } (\text{abs}(N - E)) < 3$$

$$= 3 / \text{abs}(N - E) \quad \text{otherwise.}$$

The score array is smoothed with a moving average filter (window = 5) and a hit is predicted where the score exceeds a user defined threshold for a number of consecutive frames within a user defined minimum and maximum duration.

2.2.10 Wind: AED and Diagonal Linear Classifier

Wind and rain are frequent acoustic 'contaminants' of environmental recordings. There are two reasons why automated recognition of these episodes might be useful. First, in most cases the user will want to minimise storage and computation by avoiding wind and rain events that mask useful information. Second, although wind and rain can be detected using meteorological instruments, hardware security is a problem in many locations—indirect evidence of wind and rain could help to interpret other features of a recording. While foam baffles can be used to cover microphones and reduce the effect of wind, in practice we have found that once wet they retain moisture.

We approached wind detection as a classification task, where the entities being classified are acoustic events extracted using AED as described above. Gusting wind events (e.g. Figure 1(k)) are found in the low frequency range and AED is able to marquee those that would hinder further analysis. We explored a range of event features and adopted four; two describing the distribution of acoustic intensity and two describing acoustic entropy.

Wind Feature 1: Step 1: Calculate the mean pixel intensity for each frequency bin in the event. Step 2: Calculate the difference, in decibels, between two mean intensity values—the maximum value for any bin located below 500Hz and the minimum value

in those bins above the maximum bin (see Figure 3(b)). The intensity difference is greater for wind events.

Wind Feature 2: Calculate the difference, in Hertz, between two frequency values—the frequency at which the maximum mean intensity is located and the frequency at which the minimum mean intensity is located (Figure 3(b)). The frequency difference is greater for wind events.

Wind Feature 3: Step 1: Calculate the entropy of the pixel intensity values in each frequency bin. Step 2: Calculate the difference between two entropy values—the minimum entropy value for any frequency bin located below 500Hz and the maximum entropy value for any bin above the location of the minimum (see Figure 3(c)). The entropy difference is greater for wind events.

Wind Feature 4: Calculate the difference, in Hertz, between two frequency values—the frequency at which the maximum entropy is located and that at which the minimum entropy is located (Figure 3(c)). The frequency difference is greater for wind events.

For training and test data we extracted all events whose minimum and maximum frequencies were <500 Hz and <2 kHz respectively. Events had to be longer than one second for reliable tagging but the extracted features are duration independent and therefore the trained classifier is able to label events shorter than one second.

2.2.11 Rain: AED and Linear Classifier

As with wind detection, we approached rain detection as a task to classify AED events. The selected rain events consisted of heavy canopy rain. In particular, they excluded light rain and drizzle. During canopy rain, broadband percussive effects arise from the striking of large rain drops on surfaces near the microphone (e.g., Figure 4(a)). The following three features were found to offer reasonable detection accuracy. Two describe acoustic intensity and the third describes acoustic entropy.

Rain Feature 1: The selected canopy rain events included discernible raindrops which left a trace of about ten intensity peaks per second (Figure 4(b)). Feature 1 is the mean interval in seconds between the intensity peaks.

Rain Feature 2: The difference between the minimum and maximum raindrop interval Figure 4(b). The difference is less for canopy rain events.

Rain Feature 3: Observation revealed that canopy rain events have higher entropy values in the 8.5-10.5 kHz frequency bins than do non-rain events. Therefore feature 3 is the mean entropy value of frequency bins in the range 8.5–10.5 kHz (Figure 4(c)).

For training and test data we extracted all events whose minimum and maximum frequencies were >1 kHz and >8.5 kHz respectively and whose bandwidth was greater than 2 kHz. Reliable tagging required events of duration longer than three seconds but once again, the features are duration independent (for events longer than 0.5 seconds) and therefore the trained classifier can be used to label events shorter than 3 seconds.

3 Experiments

In this section we demonstrate the above techniques on a variety of animal calls that are representative of those of interest to the ecologists with whom we are working. We describe the hardware, data preparation and the recognition accuracy for the various techniques.

3.1 Hardware

3.1.1 Networked Sensors

These devices provide a near real-time sensing in locations with 3G connectivity. Due to bandwidth constraints and 3G transmission costs, they are configured to activate at regular intervals (typically two to four minutes at half hourly intervals), upload data to a central repository and then deactivate until the next scheduled recording. These networked sensors consist of a 3G smartphone (HTC), an electret-style external microphone, pre-amplifier, DC-DC converter and an external power supply (solar panel). The system is capable of operating continuously and autonomously for months at a time with minimal maintenance. Uninterrupted, continuous deployments of 18 months have been achieved to date (Wimmer et al. 2010).

3.1.2 Acoustic Loggers

Acoustic loggers are designed to provide a short term or ad hoc high resolution acoustic sensing capability but can be configured to record continuously for extended periods (Wimmer et al. 2010). Recorded acoustic data is stored internally and the device provides up to 14 days continuous recording. These sensors are highly portable

and powered by batteries for ad hoc deployments or externally using a solar power supply for fixed deployments. As with the networked devices, these sensors are self-contained in an all-weather container.

3.2 Experimental Design

All recordings were obtained from various locations on the east coast of Queensland, Australia, using either networked sensors or data loggers. Most of the recordings were obtained in the context of three research projects, one on koalas (FitzGibbon et al. 2009), another on quolls (Belcher et al. 2008) and another at a university field station (Williamson et al. 2008).

Networked sensors returned recordings of two or four minutes depending on the protocol requested by the ecologist. Data loggers returned recordings of several hours duration. These were split into lengths of two to four minutes.

Much consideration needs to be given to the reporting of recognizer accuracy. In the absence of generally accepted standardised datasets, recognition accuracy can be made arbitrarily high depending on the selection and prior cleaning of the recordings. We have endeavoured in these experiments to construct datasets that reflect the 'real-world' of sound which ecologists must process. Datasets were chosen according to their expected ecological significance rather than to obtain clean recordings. For example, recordings containing wind and rain 'contaminants' were not removed. Some of the recordings include the morning and evening chorus where there can be a cacophony of sound. Others contain traffic noises, air-conditioners (in city recordings), human speech, dogs barking and airplanes. Finally the tagging of calls

was done by ecologists whose trained ears could detect faint distant calls that one would not expect to detect by automated means. Only in this last case did we exclude from consideration calls whose intensity was less than 2 - 4 dB above the background noise level.

Another issue in the context of our data was whether to express accuracy in terms of correctly identified calls or correctly identified recordings. We opted for the latter because the principle cost of an error is the time spent online by an ecologist loading a file and accessing the predicted call. Consequently we use the following standard definitions for recall and precision:

$$\text{Recall} = TP/(TP+FN),$$

and

$$\text{Precision} = TP/(TP+FP),$$

but define TP (true positives) as the number of short (2-4 minute) recordings correctly identified as containing one or more calls of interest; FN (false negatives) as the number of files containing positive calls none of which were identified; and FP (false positives) as the number of files incorrectly identified as containing one or more calls.

Scoring on the basis of recordings has two effects on the reported accuracy. Where a recogniser detects a TP yet makes an FP or FN error in the same file, we label that file correctly classified. On the other hand we observed many instances where multiple TPs obtained in one recording were offset by a single error in another file. This situation arises because many bird calls appear in temporal clusters. In short, the accuracy figures presented below must therefore be regarded as indications of operational performance in a real problem as opposed to finely tuned estimates of

accuracy. Note that all the experiments were designed to produce a binary output – call identified/not identified.

3.3 Calls

3.3.1 Currawong (*Strepera graculina*)

The currawong dataset consisted of 29 four-minute recordings taken at half-hourly intervals from midnight to 7am and 5pm to midnight on the 2nd November 2009 at St Bees Island. This protocol was designed to encompass both sides of the morning and evening chorus. The dominant calling species is the curlew. However 5 of the 29 recordings contained currawong calls and typically their calls are clustered suggesting flocks of the birds making an appearance during the recording.

We compared two recognition techniques, HMMs trained on MFCC features and a simple binary template. As can be seen from the results in Table I, neither method performed well. However there are two points to note here: First much time was spent obtaining training and validation data for the HMM method while the binary template was quickly prepared from only one representative call, manually edited by the user to remove artefacts and echo. Second, the currawong calls were numerous but clustered into just five files. The binary template recognised the great majority of the calls and therefore on a call basis its recall would have far exceeded that of the HMM. The low recall of the HMM approach was due to the difficulty of training a suitable noise model that covered the range of ambient noise situations. The low precision of the binary template was primarily due to false positive identifications of the more numerous curlew calls which sit in the same frequency band.

3.3.2 Beach Stone-curlew (*Esacus neglectus*)

The curlew dataset was the same as that used to test the currawong recognizer. 19 of the 29 recordings contain curlew calls, although 9 of them are of low intensity due to distance of the source. The curlew produces a variety of syllable types during one call but the syllable shown in Figure 1(b) is sufficiently well defined to warrant targeting. We employed a binary template and achieved an accuracy of 76% (Table I). The dominant errors were false negatives due to the large proportion of low intensity calls.

3.3.3 Male Koala (*Phascolarctos cinereus*)

The male koala bellow consists of a sequence of loud inhalations and exhalations. The exhalations have an oscillatory component which offers a suitable target for recognition. Training data (for tuning the OD parameters) consisted of 12 four-minute recordings, each containing a koala call. Test data, totalling 7 hours 40 mins (460 minutes), were split into 115 four-minute recordings, 12 of which contained koala calls. There were 18 calls, ranging from high to low intensity. This selection of recordings having low call density is representative of the real world situation.

Recall was 75% with the three false negative files containing distant bellows of very low intensity. To measure precision, we took into account that koala bellows contain multiple oscillatory exhalations which will be detected in clusters. Consequently files containing only a single hit are likely to be false positives. Ignoring recordings with fewer than two hits resulted in a precision of 75%. False positives were mostly due to a bird (the Orange Footed Scrub Fowl) with a deep, chattering call, producing oscillations in a frequency band overlapping that of the male koala. Although

precision and recall give some indication of accuracy of the detection, these metrics ignore the large volume of data that was scanned to get the result. Based on total files correctly classified, we obtain an accuracy of 95%. For the ecologist this is a large saving in time.

3.3.4 Cane Toad (*Bufo marinus*)

Cane toad data was collected in a suburban backyard (Brisbane, Queensland, January 2010) and in rural farmland (near Gympie, South East Queensland). Both locations were in the vicinity of permanent or semi-permanent water bodies with known cane toad populations. A total of 674 minutes of recording were split into 337 two-minute files. The dataset contained 83 cane toad calls in 53 files. The suburban recordings were 'contaminated' with a wide variety of extraneous sounds including traffic, air-conditioning, speech, dogs etc.

The OD recogniser achieved a recall and precision of 93% and 98% respectively, and an accuracy of 99% over 337 files. These high accuracy rates are partly due to the fact that cane toads have very consistent call characteristics which make it possible to fine tune the recogniser parameters. The false positives fall into two categories: kookaburra calls that oscillate in same frequency range as cane toads and very short hits due to background noise. The false negatives were all low intensity calls. Additionally, one call was masked by wind, an unwelcome contaminant that affects the accuracy of animal call recognition in all real world tasks.

3.3.5 Asian House Gecko (*Hemidactylus frenatus*)

Asian house gecko recordings were recorded in January and February 2010 just outside a suburban house (Brisbane, Queensland). 540 minutes of recording were split into 270 two-minute files, 77 of which contained 84 calls. As shown in Figure 1(e), the gecko call consists of a succession of chirps which can be detected using the OD algorithm. Recall and precision were 91% and 90% respectively with an accuracy of 94% over the 270 files. The high accuracies confirm that geckos, like cane toads, have very consistent call characteristics that are readily detectible by the OD recogniser. False negatives errors were due to missing calls of low intensity.

3.3.6 Segmentation for OD Recognition Experiments

Recognition efficiency can be greatly increased by ignoring those parts of a recording in which there is no acoustic activity in the bandwidth of interest. We repeated the recognition experiments for male koalas, cane toads and geckos using a segmentation algorithm to remove silence.

Table II confirms that high accuracy rates can be retained when using the segmentation filter prior to OD recognition. In the case of the gecko, accuracy actually increased due to the filtering out of false positives ('noise'). However in the case of the cane toad data, segmentation removed some weak calls thus lowering recall. The main purpose of the segmentation filter however was to reduce processing time and in this it was successful - up to 87% time reduction in the case of the gecko data (Table II, bottom right cell).

3.3.7 Ground Parrot (*Pezoporus wallicus*)

Ground parrot data was collected with data loggers placed in a nature reserve 100 km north of Brisbane. We acquired a total of 6 hours and 45 minutes of recordings, which were broken into one-minute sections. An ecologist tagged the calls (see Figure 1(f)). Of the 405 one-minute files, 32 contained calls. However nine of these were barely audible. It should be noted that 1-2 dB is the minimum perceptible audible difference between signal and background noise (Lüscher 1951). Consequently ground parrot calls whose maximum intensity was less than 2 dB above background noise were ignored for testing purposes.

We used EPR for detecting a specific type of ground parrot call, a stable pattern of short-duration, narrow-band, faint chirps that ascend in the frequency as shown in Figure 1(e) and Figure 2. Note that ground parrots have other vocalisations that differ from our call of interest.

AED's two default parameters were modified: the intensity threshold was reduced from 9 to 3 dB (because many of the calls were faint) and the small area threshold was lowered from 200 to 100 pixels (because the component syllables are small). The EPR algorithm overlays the ground parrot template (see Figure 2) on acoustic events detected by AED. To limit unnecessary computation, only events were considered whose centroid lay within the 3.5 - 4.5 kHz bandwidth of interest. For each location, the bottom-left vertex of the template was aligned to the bottom-left vertex of the selected 'test' event. The precision and recall rates cited below were calculated using a threshold overlap of 27% determined on the total data. There was not enough data for a separate validation and test set.

Precision and recall for the 23 files with audible parrot calls were both 87%. Two of the three false negatives detected were faint calls (compared to the true positives), and the other was missed because of edge effects (i.e. only half a call was present at the end of a recording). Three false positives resulted from incorrect detections due to rain which presented in the spectrogram as a random distribution of acoustic events. Total accuracy was 99%. We believe this is a particularly effective technique for multi-syllable calls which have a characteristic distribution of events in the spectrogram.

3.3.8 Eastern Whipbird (*Psophodes olivaceus*)

The whipbird has a two syllable call consisting of a whistle followed by a whip (Figure 1(g)). It is a good candidate for the Syntactic Pattern Recognition approach. The dataset consists of 38 two minute recordings, 14 of which contain whipbird calls. The accuracy of 82% shown in Table I is somewhat misleading in that whip bird calls are clustered and some of the recordings contained many true positive recognitions that counted only as one TP on a file basis.

3.3.9 Torresian Crow (*Corvus orru*)

The crow spectrogram has a set of stacked harmonics that extend up into the 8 kHz range depending on how close the bird is to the microphone. It is clearly a candidate for our HD algorithm. The crow dataset consisted of 20 four-minute recordings, 12 of which contained crow calls. Recall was 100% and precision was 71% with an accuracy of 75%. We have used the same algorithm (with different parameter values)

for recognition of female koala calls and the vowels in human speech but we do not present results due to lack of suitable data in our environmental recordings.

3.3.10 Wind Events

A classifier was trained with 142 'wind' and 142 'not-wind' events using Matlab's classify.m class. 'Not-wind' events included low frequency rumbling due to traffic and aircraft. Best results on a validation set of 383 'wind' events and 243 'not-wind' events were obtained with a diagonal-linear classifier. The training and validation error rates were 10% and 14% respectively.

We tested the diagonal-linear classifier on 1235 one-minute audio files (Table I), acquired with a variety of sensors at different locations. The recordings included koala bellows, ground parrot calls and many other kinds of acoustic events. The accuracy rate was 96% on a one-minute recording basis (Table I). False positive detections were due to male koala bellows that have low frequency content. The FP-FN trade-off for the wind detector could be adjusted using a constant which shifts the decision plane towards one class or the other. We used the default value of zero for this constant.

3.3.11 Rain Events

Canopy rain events were extracted from recordings taken at five different locations in Queensland, Australia. The events were classified as 'rain', 'not-rain' or 'not-sure'; events needed to be longer than 3 seconds for reliable classification but even so, many events were classed as 'not-sure'. The training and validation sets excluded 'not-sure'

events and consequently there was less data available than for the wind classifier. Non-rain events primarily consisted of percussive sounds resulting from construction and human activity. A classifier was trained with 54 'rain' and 52 'not-rain' events using Matlab's classify.m function. A validation set of 19 'rain' and 14 'not-rain' events was used to determine that a Linear classifier provided best results. We tested the classifier on 247 one-minute files, 104 of which contained rain events and achieved recall, precision and accuracy rates of 75%, 75% and 79% respectively (Table I).

It is worth noting that rain detection is the more difficult than wind detection, both for humans and for the machine classifier. Indeed, the poorer performance of the rain classifier was probably in part due to the prior inaccurate tagging of events that were used to train the classifier.

4 Discussion and Conclusion

In this work we have described a toolbox of call recognition techniques to detect animal calls in environmental recordings. Our objective was to report performance figures for experimental conditions that reflect the needs of ecologists having to process many hours of recordings. The work reported here arose out of an early realisation that a one-recogniser-fits-all approach would not cope with the unconstrained variety of acoustic events that appear in environmental recordings. In particular, the highly refined techniques of ASR using MFCCs and HMMs have been tailored to a very constrained audio environment and, even on theoretical grounds, they are ill-suited to unconstrained audio environments. Of course ASR techniques

can be made to perform well on datasets of animal calls that have been clipped from recordings to make a balanced N-class problem. But this is not the real world of ecology.

Our recognisers have a number of features that reflect real world usage:

1. With the exception of the MFCC-HMM approach, our recognisers can be trained in the first instance with just one or very few training instances. The recognisers can be refined when new instances become available. This is necessary because most often large numbers of training calls are not available and large training and tests sets are time consuming to curate.
2. The recognisers are constructed as binary classifiers, i.e. they detect presence or absence. This is in contrast to an *N*-class classifier which must assign new instances to the most likely class. The difficulty with an *N*-class classifier is how to represent the *null* class when the audio content is unconstrained.
3. An additional difficulty with the single *N*-class classifier is that typically the discovery of a new instance or class requires the retraining of the entire classifier. With *N* binary classifiers, only one of them needs retraining.
4. In our experience successful classification depends more on extracting the appropriate features for a task than on the sophistication of the classifier. With the one-recogniser-fits-all approach the same feature set must be extracted for all calls and consequently a powerful classifier is required because it is difficult to separate the classes in feature space. By contrast, given a set of appropriate features, a linear classifier is often good enough. This was most obviously demonstrated in the wind and rain classification tasks where a linear

classifier was the best (of those offered by Matlab) at separating the two classes.

5. For the most part our binary classifiers have tuning parameters whose function is intuitively clear for the non-technical ecologist. For example, calls have a minimum and a maximum duration and amplitude thresholds are represented in decibels. The obvious exception is the MFCC-HMM approach whose many parameters require a basic understanding of the theory.

There are two obvious extensions to our work. The first is to construct classifiers for additional call types. The second is more challenging. At the present time our recognisers return a score either in dB units or normalised in [0, 1]. Therefore we are not able to make a choice between two simultaneous 'hits' if the score is in different units. We intend to normalise all scores so that it is possible to disambiguate cases where a single acoustic event returns positive to more than one classifier. This effectively offers the possibility of using the *N* binary classifiers to imitate an *N*-class nearest neighbour classifier.

Acknowledgement

The Microsoft QUT eResearch Centre is funded by the Queensland State Government under a Smart State Innovation Fund (National and International Research Alliances Program), Microsoft Research and QUT.

Tables

Table I. Toolbox for Call Recognition

Generic Call Structure	Call	Features Extracted from Spectrograms	Classification Algorithm	Recordings (Files in Datasets)	# Files with Calls	Recall	Precision	Accuracy
Frequency-modulated whistle(s)	Currawong	MFCC	HMM	29 x 4-minutes	5	40%	100%	90%
		Binary template	Template matching			100%	50%	83%
	Curlew	Binary template	Template matching	29 x 4-minutes	19	63%	100%	76%
Oscillatory components	Male Koala	Oscillation rate and amplitude using DCT.	Threshold classifier	115 x 4-minutes	12	75%	75%	95%
	Cane Toad			337 x 2-minutes	55	93%	98%	99%
	Asian House Gecko			270 x 2-minutes	77	91%	90%	94%
Syllable events having fixed distribution in spectrogram	Ground Parrot	AED	Event Pattern Recognition	405 x 1-minute	23	87%	87%	99%
Combinations of whistles, chirps, whips	Whipbird	Orientation of spectral tracks	Syntactic Pattern Recognition	38 x 2-mintues	14	100%	67%	82%
Stacked harmonics	Crow	Frequency spacing of parallel spectral tracks.	Threshold classifier	20 x 4-minutes	15	100%	71%	75%
Diffuse	Wind	AED/Image features	Diag. linear classifier	1235 x 1-minute	792	99%	96%	96%
	Rain	AED/Image features	Linear classifier	247 x 1-minute	104	75%	75%	79%

Table II. OD Parameter Settings and Recognition Results

	Dataset		
	Male koala	*Cane toad*	*Asian house gecko*
Parameter			
Frequency band	0.1 – 1.0 kHz	0.5 – 1.0 kHz	1.5 – 3.0 kHz
Frame overlap	75%	75%	0%
DCT duration	0.3 sec	0.5 sec	1.0 sec
Oscillation bounds	20 – 50 Hz	10 – 20 Hz	3 – 7 Hz*
Min DCT amplitude	0.6	0.6	0.5
Call duration	0.5 – 2.5 sec	0.5 – 20.0 sec	1.0 – 6.0 sec
% of freq. bins with oscillations	20%	40%	30%
OD Recognition Results			
Total Number of Files	115	337	270
True Positives Files	9	49	70
False Positives Files	3	1	8
False Negatives Files	3	4	7
Recall	75%	93%	91%
Precision	75%	98%	90%
Accuracy	95%	99%	94%
OD Recognition Results - With Segmentation			
Recall	75%	83%	91%
Precision	75%	98%	93%
Accuracy	95%	97%	96%
Processing Time (% of the original time w/o segmentation)	34%	56%	13%

*Asian house gecko oscillation bounds were derived based on behavioural studies described in (Marcellini 1974).

Figure Captions

Figure 1. Examples of animal vocalisations and other sounds.

Figure 2. Example Ground Parrot template.

Figure 3. Wind image features.

Figure 4. Rain image features.

References

Acevedo, M. A., C. J. Corrada-Bravo, H. Corrada-Bravo, L. J. Villanueva-Rivera, and T. M. Aide. 2009. Automated classification of bird and amphibian calls using machine learning: A comparison of methods. Ecological Informatics **4**:206-214.

Agranat, I. 2009. Automatically Identifying Animal Species from their Vocalizations. Fifth International Conference on Bio-Acoustics, Holywell Park.

Akhtar, M., E. Ambikairajah, and J. Epps. 2007. GMM-Based Classification of Genomic Sequences. Pages 103 - 106. The International Conference on Digital Signal Processing.

Anderson, S., A. Dave, and D. Margoliash. 1996. Template-based automatic recognition of birdsong syllables from continuous recordings. Journal of the Acoustical Society of America **100**:1209-1219.

Belcher, C., M. Jones, and S. Burnett 2008. Spotted-tailed quoll, Dasyurus maculatus. New Holland Publishers.

Brandes, S. T. 2008. Automated sound recording and analysis techniques for bird surveys and conservation. Bird Conservation International **18**:S163-S173.

Brandes, T., P. Naskrecki, and H. Figueroa. 2006. Using image processing to detect and classify narrow-band cricket and frog calls. The Journal of the Acoustical Society of America **120**:2950-2957.

Brunelli, R. 2009. Template matching techniques in computer vision: theory and practice John Wiley & Sons, West Sussex.

Bunke, H., and A. Sanfeliu, editors. 1990. Syntactic and structural pattern recognition – theory and applications. World Scientific Publishing Co, Singapore.

Cai, J., D. Ee, B. Pham, P. Roe, and J. Zhang. 2007. Sensor network for the monitoring of ecosystem: Bird species recognition. Pages 293–298. Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing. Citeseer, Melbourne.

Catchpole, C., and P. Slater 1995. Bird song: Biological themes and variations. Press Syndicate University of Cambridge, Cambridge.

Chen, Z., and R. Maher. 2006. Semi-automatic classification of bird vocalizations using spectral peak tracks. The Journal of the Acoustical Society of America **120**:2974.

Cowling, M., and R. Sitte. 2003. Comparison of techniques for environmental sound recognition. Pattern Recognition Letters **24**:2895-2907.

FitzGibbon, S., W. Ellis, and F. Carrick. 2009. Mines, farms, koalas and GPS-loggers: assessing the ecological value of riparian vegetation in central Queensland. Page Poster Presentation. The 10th International Congress of Ecology.

Juang, C., and T. Chen. 2007. Birdsong recognition using prediction-based recurrent neural fuzzy networks. Neurocomputing **71**:121-130.

Kwan, C., G. Mei, X. Zhao, Z. Ren, R. Xu, V. Stanford, C. Rochet, J. Aube, and K. C. Ho. 2004. Bird classification algorithms: theory and experimental results. Pages V-289-292 vol.285. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).

Lau, A., R. Mason, B. Pham, M. Richards, P. Roe, and J. Zhang. 2008. Monitoring the environment through acoustics using smartphone-based sensors and 3G networking in K. Langendoen, editor. Proceedings of the Second International Workshop on Wireless Sensor Network Deployments (WiDeploy08); 4th

IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2008. IEEE.

Lüscher, E. 1951. The Difference Limen of Intensity Variations of Pure Tones and its Diagnostic Significance. The Journal of Laryngology & Otology **65**:486-510.

Marcellini, D. L. 1974. Acoustic Behavior of the Gekkonid Lizard, Hemidactylus frenatus. Herpetologica **30**:44-52.

McIlraith, A. L., and H. C. Card. 1997. Birdsong recognition using backpropagation and multivariate statistics. IEEE Transactions on Signal Processing **45**:2740-2748.

Rickwood, P., and A. Taylor. 2008. Methods for automatically analyzing humpback song units. Journal of the Acoustical Society of America **123**:1763-1772.

Somervuo, P., A. Harma, and S. Fagerlund. 2006. Parametric Representations of Bird Sounds for Automatic Species Recognition. IEEE Transactions on Audio, Speech, and Language Processing **14**:2252-2263.

Temko, A., R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo. 2006. Acoustic event detection and classification in smart-room environments: Evaluation of CHIL project systems. Cough **65**:5-11.

Towsey, M., and B. Planitz. 2010. **Technical Report: Acoustic analysis of the natural environment**. Queensland University of Technology, Brisbane.

Towsey, M., B. Planitz, J. Wimmer, and P. Roe. 2010. Animal Call Recognition Using Oscillation Detection. Page (submitted). The 2nd Conference on Environmental Science and Information Application Technology (ISSNIP 2010).

Trifa, V., A. Kirschel, C. Taylor, and E. Vallejo. 2008. Automated species recognition of antbirds in a Mexican rainforest using hidden Markov models. The Journal of the Acoustical Society of America **123**:2424-2431.

Williamson, I., S. Fuller, and C. Marston. 2008. A Vertebrate Survey of the Samford Ecological Research Facility. School of Natural Resource Sciences, Queensland University of Technology (QUT), Brisbane.

Wimmer, J., M. Towsey, B. Planitz, and P. Roe. 2010. Scaling Acoustic Data Analysis through Collaboration and Automation. Page (submitted). IEEE eScience 2010 Conference, Brisbane, Australia.

Young, S., G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland 2006. The HTK book (for HTK Version 3.4). Cambridge University Engineering Dept., Cambridge.

Zhuang, X., Z. Xi, T. S. Huang, and M. Hasegawa-Johnson. 2008. Feature analysis and selection for acoustic event detection. Pages 17-20. Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.