Improvements in an automatic sound recognition system using multiple parameters to permit recognition with noisy and complex signals such as the dawn chorus

Neil Boucher, Michihiro Jinnai, Andrew Smolders

ACOUSTICS 2012

Improvements in an automatic sound recognition system using multiple parameters to permit recognition with noisy and complex signals such as the dawn chorus

N. Boucher[a], M. Jinnai[b] and A. Smolders[c]

[a]SoundID, PO Box 649 Maleny, 4552 Queensland, Australia
[b]Kagawa National College of Technology, 355 Chokushi-cho, 761-8058 Takamatsu, Japan
[c]University of New England, Armidale, 2350 Armidale, Australia
nboucher@ozemail.com.au

Improvements in an automatic sound recognition system using multiple parameters to permit recognition with noisy and complex signals such as the dawn chorus are described. We show that with a suitable selection of parameters it is possible to work with very noisy signals, and with some limitations down to -20 dB S/N. Because our technique is highly mathematical, we show that it is relatively easy to trade CPU time for accuracy. Alternatively we can trade accuracy for the ability to work in very high noise environment significantly below 0 dB. We describe a system that can challenge a human expert at any sound recognition task.

1 Introduction

When we began the study of sound recognition we naturally started with high quality signals that had very low levels of noise. The approach was to transform a sound into an image generated using the LPC frequency transform (frequency vs energy within a given frame-width) and compare that image to a library of images that comprise the reference files. The sounds were recorded by professional wild-life sound recordists. They will ordinarily have a noise floor of 50 dB or better and the signal will span most of the 22050 Hz range. Typically an image will be as seen in Figure 1 below.

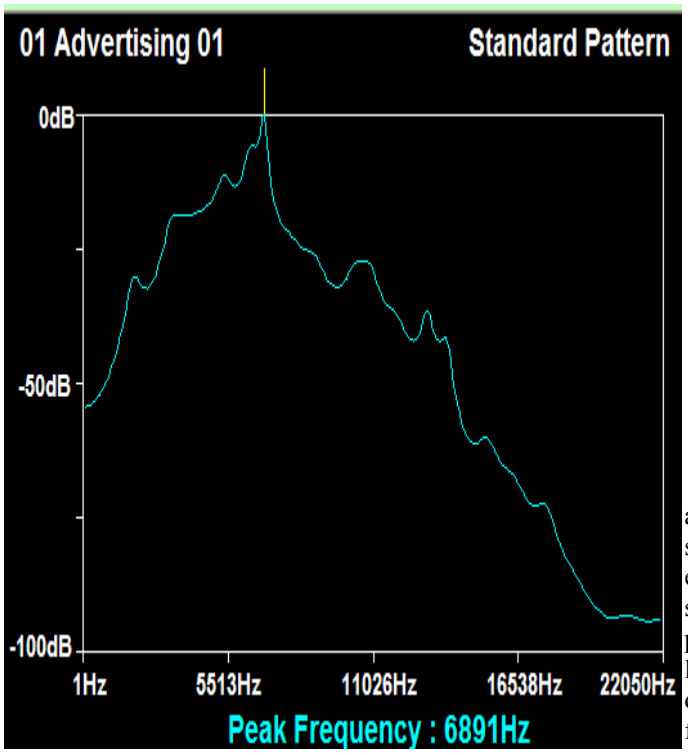

Figure 1 A typical professional sound image.

After some development work we were able to achieve very high quality recognition of such signals (100% accuracy with 0% false positives). When we moved from the development platform to the real world the situation was somewhat different. It was found that such "clean" signals as the one above are very rare and mostly the signal was competing with other sounds and was buried in a significant noise floor. In this paper we discuss the techniques that were developed to enable good recognition in a noisy environment. In general, because our technique is entirely mathematical we can sacrifice CPU time for accuracy and this is increasingly what we are doing.

2 Noise Characterisation

To begin, we noted that the Signal to Noise ratio (S/N) that is generally accepted as the limiting value for effective voice communications in an analogue system is 10 dB and so we began by looking at signals of that level. The same signal (as the one in Figure 1) with 10 dB S/N is shown in Figure 2.

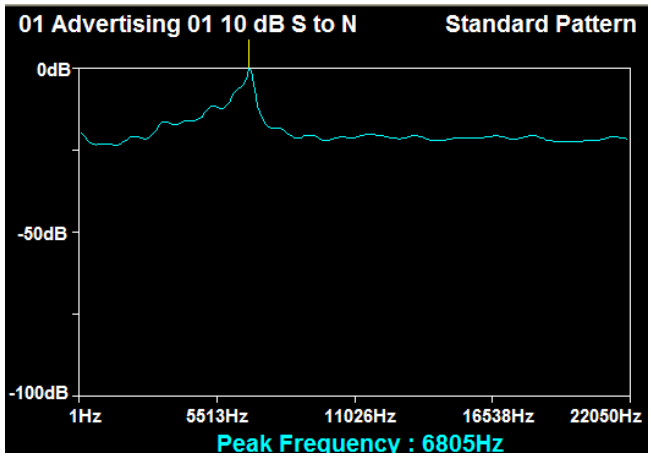

Figure 2. The signal from Figure 1 in S/N of 10 dB.

It is easy to see that if we are to compare these images and ask the question are they the same, or even rather similar the answer is definitely no! So we really can't expect the software, that is simply image matching to see a similarity. Intuitively one might suggest that if noise is a problem, then simply use noisy references. This fails on at least two levels, firstly that the image is very different for different S/N ratios, so we would need a lot of references for each signal and secondly the noise is itself not of a consistent nature, and so we have references that might lead to noise matching with noise. The noise added in Figure 2 is pink noise.

3 Comparing Signals

If we compare the two images of Figure 1 and Figure 2 using the software we get the result seen in Figure 3. Because changing parameters also affects run-time (which may be significant for very large runs), we report a batch run-time figure in seconds. This is the time taken by this particular software to compare 85 sound images. The actual time is not important, what is, is the relative time. So we need to keep in mind that mostly anything that increases the precision of the recognition also increases the CPU time to run it. Because the system is designed to be able to run

terabytes of data, even small increases in run-time can ultimately be an issue of some consequence.

Figure 3 shows not only the images that are matched, but the parameters that we can optimise to get the best match (at the bottom of Figure 3). These are the frame width (number of points), LPC order, processing bandwidth (F1 and F2) and LPC depth (set to be just above the effective noise floor). We call a reference file, with its settings (all saved in the one file) a template.

When the images are compared the software will return a Geometric Distance (GD) giving the closeness of the match. The GD is a similarity measure developed by one of us (Jinnai). In Figure 3 we see the GD between the "clean" signal and the noisy one in a purple text box and reading 10.84. This indicates, as would be expected that the two images are very dissimilar. A GD of six or less would be required to indicate a good matching similarity.

So what is needed is a way to preserve similarity between the "clean" reference calls and their counterparts in the real world that is noisy and competing with other sounds. We cannot assume that we have prior knowledge of the nature of the noise except of course for the 1/f noise that is largely due to the wind. Provided the targets are not calling at frequencies similar to the wind noise filters can be used either at the time of the recording or post-recording.

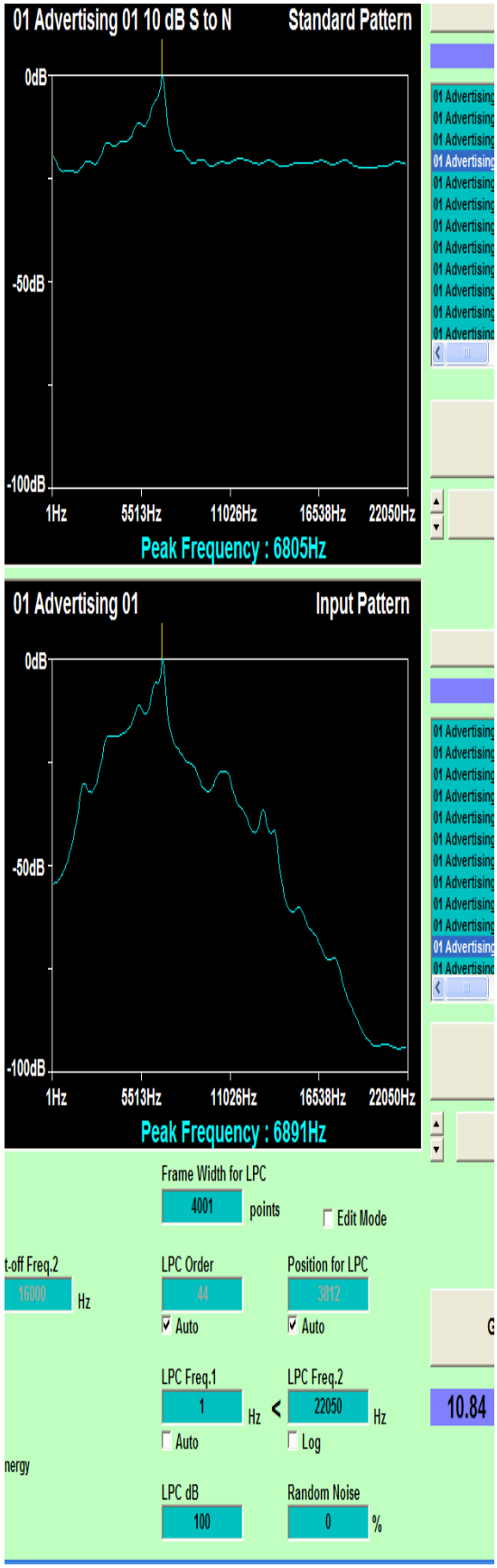

Figure 3. The noisy signal matches with the "clean" one with a GD=10.84 (a very poor match). Batch time = 9 seconds

3.1 Gate the Problem

Look closely at Figure 2 and you can see some very salient points. Firstly the noise floor is at about 25 dB and there is very little information at levels below that. Also we

can see that the image has no recognizable parts (the noise floor is not part of the signal) much below 2500 Hz of above 8000 Hz. So one of us, Jinnai, devised a gating solution to this. The gate is a frame built around a portion of the signal that we want to concentrate on. In figure 4 below we see the recognizable part of the noisy signal gated, as described above.

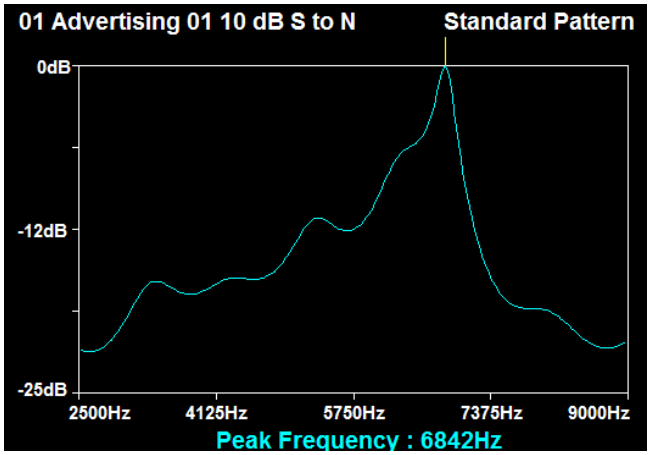

Figure 4 The gated noisy signal.

To see that this approach works compare Figure 4 (the gated noisy signal) with Figure 5 (the gated "clean" reference). The similarity has been restored. The one thing that spoils the image is the clipping seen in Figure 5 at the extremities of the frequency spanned. This suggests gating the frequency from 3000 Hz to 8000 Hz.

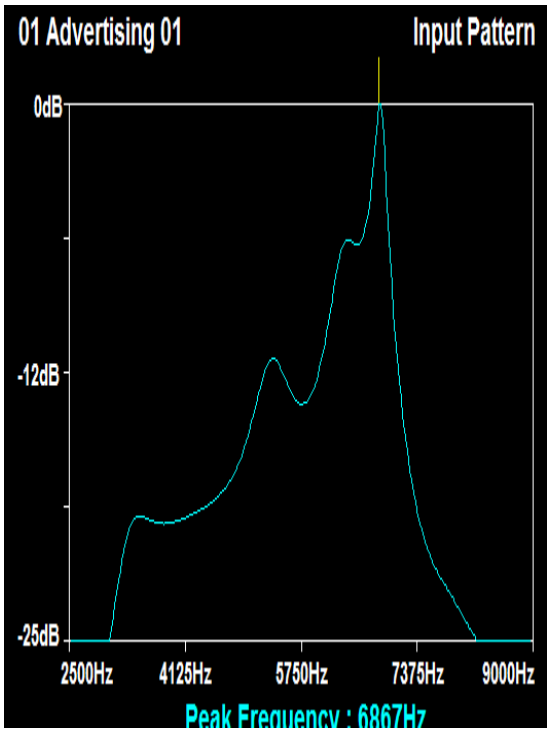

Figure 5. The gated clean signal.

In figure 6 below we see the gated signal compared and that they match to a level of GD=4.90 (good match). Here we need to comment a bit more on the GD. The way we use it, it has units of degrees (angular degrees) where 0 degree = a perfect match and 90 degrees = no possible match. To a first approximation (and only as an analogy) it is reasonable to think of the matching as logarithmic. So a match of GD =4.9 compared to a match of GD =10.84 are $10^{(10.84-4.90)} = 10^{5.94}$ apart; that is, there is a huge difference in the match. If 4.9 is a good match then 10.84 is close to no match at all. In general we would say that for a gated signal a GD of less that 6.00 degrees is a good match and that anything higher is a non-match.

So we have made some significant progress by gating the signal in such a way that we are focusing on the part of the signal least affected by the noise.

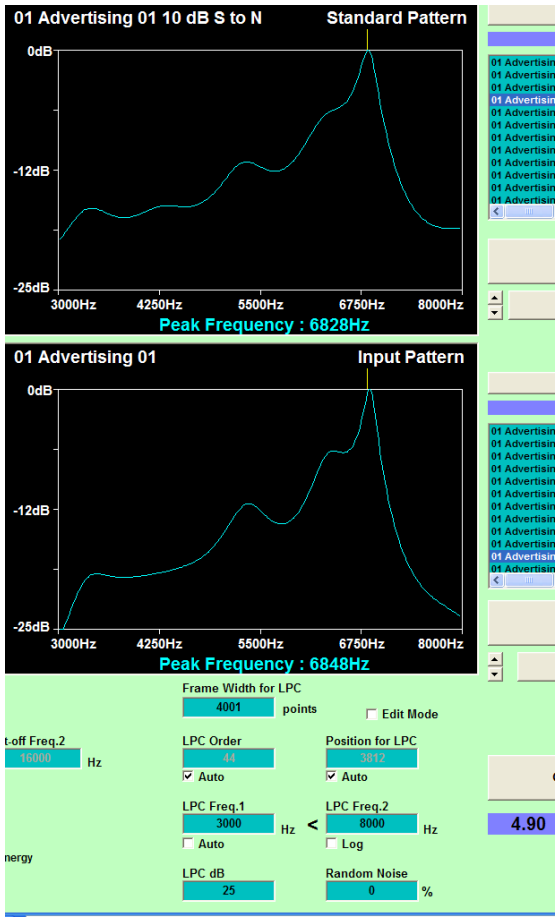

Figure 6. A comparision of the gated figures. Batch time= 9 seconds.

The gating approach so far has improved the matching and focusing on the most energetic part of the signal has improved the matching without increasing the computational overhead. However we now consider an approach that will further enhance the noise performance but at considerable cost computationally. If we use a higher order LPC calculation we will extract more identity information from the signal and improve the match. But a considerable increase in CPU time will be the price to be paid for this. In Figure 7 below we have done that and can now see that the match has improved significantly to GD= 2.74 degrees. Notice we have narrowing the frequency bandwidth even more and the high order LPC will also "enhance" the noise recognition by characterizing it better.

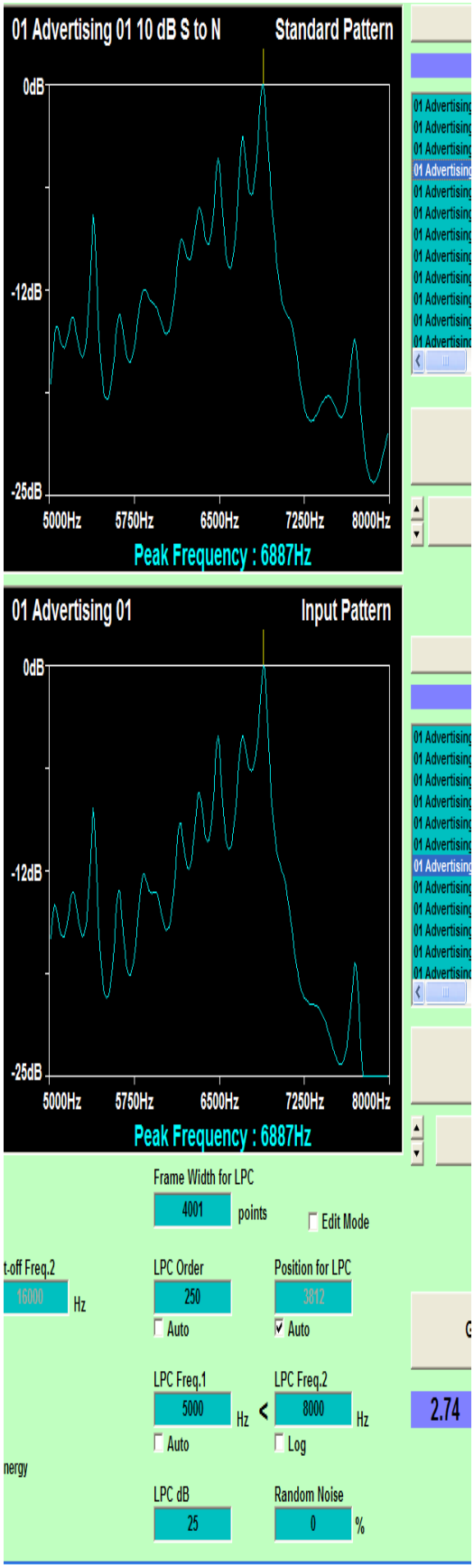

Figure 7. The same signal with the LPC order increased from the default of 44 to 250 (batch time =33 seconds).

The CPU time has increase 33/9=3.7 fold. That extra processing time is quite significant and is a real cost of the higher resolution. Here it should be noted that there is a limit to the LPC order (and hence the improvement we can get by increasing it), as a characteristic of the LPC transform is that it is unstable at high orders (typically >500), but until that point is reach it is generally true that higher orders mean better resolution.

But we still have more tricks. The software always operates on the most energetic part of the signal. That is, once the frame size is defined, the software looks for the highest energy section of the signal within the defined frame size. Hence it would seem reasonable that decreasing the frame size would focus even more on the highest energy portion of the signal and so further enhance the noise performance. As a bonus, because there will now be fewer data points to process we should recover some of the processing time. There is a limit to have far we can take this as smaller and smaller segments of the signal eventually lose some of the detail of the signal and can even cause the software to focus on local noise impulses that are not part of the signal. For the type of signal we are considering here, something around 1001 points is typically optimum.

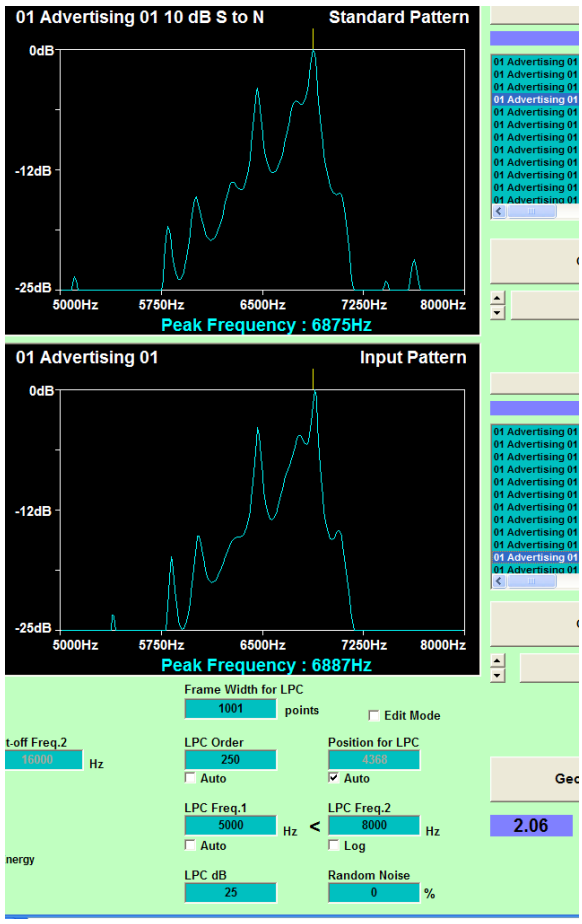

Figure 8. Running the process with 1001 points we now have a significantly improved match and reduced run-time (Batch run-time =11 seconds)

3.2 Dropping the Noise Floor Further

We have shown that it is relatively easy to work at the limit of intelligibility for humans, (10 dB S/N) but we can see that the more we focus on a small part of the signal the more of the signals character (its finger-print) is being lost. However in some instances this is a reasonable sacrifice (for example a lot of recordings of Ground Parrots in Australia are held on very noisy tracks and have S/N ratios of about -20 dB; fortunately the Ground Parrots occupy

areas that few others frequent, so that there is hardly any signal competition. And even the other species that do share the area have mostly very different calls. That being the case the software can extract the calls without seriously compromising the accuracy).

We can reduce the S/N to 0 dB and by setting the bandwidth to 6000 Hz to 7200 Hz and setting the frame-width back to 4001 points we can still get the matching to GD=2.55.

Going even further at -11.5 dB S/N and setting the bandwidth to 6300 Hz to 7100 Hz we can still get a respectable matching of GD=4.98. However if you compare the image in Figure 9 (S/N=-11.5 dB) with any of the other signal images you will see the drastically reduced detail in that image, which translates to a much higher probability of false positives. However looking at Figure 10 which compares the WAV file view of the "clean" signal with that at -11.5 dB S/N a high false positive should be expected.

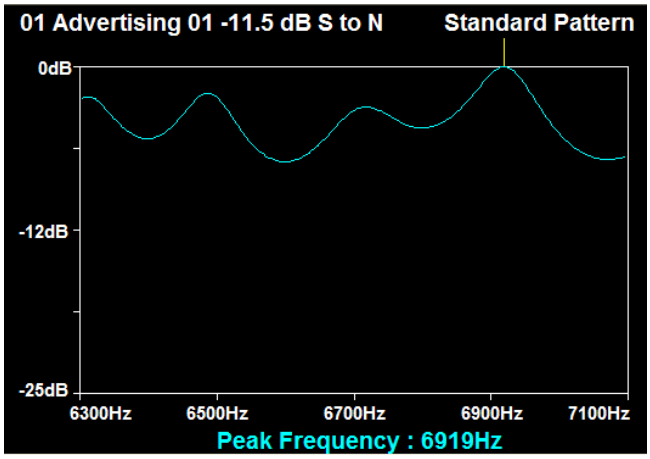

Figure 9. The image of the signal at S/N=-11.5 dB

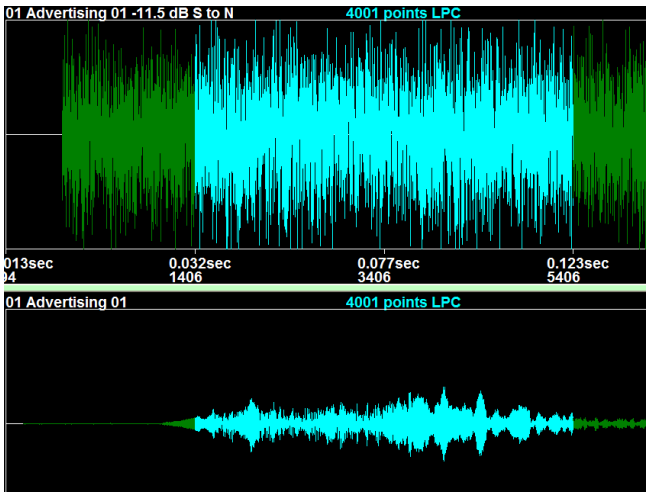

Figure 10 The wav file view of the "clean" and -11.5 S/N signals.

3.3 The Dawn Chorus

The dawn chorus is a problem for recognition systems in much the same way that noise is. There are a lot of competing calls and distant calls blend to become part of the noise background. When we first applied the techniques discussed above it was found that the recognition accuracy was high (much better than 95%), false positives very small (less than 1%), but the recognition rate was only about 600 per hour. Given that the location studied had a very active dawn chorus it was considered that the recognition rate was insufficient. With some more study it became obvious that a single template could be readily optimized for one target, but having done so, would be non-optimum for most other targets.

The solution here was to allow each target to have its own optimized template. Now we have an improvement that seriously affects the processing time. For each template the software must process the whole of the signal sequentially. So for a typical dawn chorus with 10-20 targets we need a like number of templates and we increase the processing time roughly in proportion.

But it was soon obvious that this was worthwhile. The recognition rate shot up from 600 to 10,000 to 20,000 per hour with an *increased* accuracy. The reason for the increase in accuracy is that the result of each run is effectively the same as "listening" for a particular target one at a time. Because the results are stored in a matrix that includes the time in the recording that the recognition occurred it is possible to correct false positives. If for example, two birds that can sometimes be mistaken for each are both recognized at the same point in time on their respective recognition runs then the matrix can be set to replace the less promising recognition (the one with the highest GD) with the lower one.

3.4 How Slow and Why?

The concept of trading CPU time for accuracy is a reasonable one in an era of ever increasing PC power. However for now it is a bit of a problem. Our current software is 32 bit and was fine before we introduced these latest improvements running at about 100 times faster than real-time (we could process 100 hours of recordings in 1 hour). Some of our users have terabytes of information which is many months of recordings. Running at 100 times faster than real-time still means processing a terabyte over the weekend.

But the gating and multi-reference file can be so costly in CPU time that we are down to running in real-time or slower for some things like the dawn chorus. There has already been a good deal of optimization done on the software, so there is not much chance of gains that way.

For more than a year now we have been working on a 64 bit version which promises about a 4 times increase in processing speed (for the same processes). Additionally it is much easier to utilize multiple processors in 64 bit (it is possible in 32 bit but there are limitations). So we are hopefully looking to at least 20 times faster than real time with the 64 bit version.

The 64 bit version will also run both 2 and 3 dimensional spectrograms which for some signals improves recognition (see Jinnai et al. [1]).

3.5 How Does it Compare to a Human?

Even the with now considerably slower processing which is a consequence of these new techniques, it is still much faster and more accurate than a human. The software is effectively comparing 2,000 to 5,000 spectrograms per second (it can run even faster with a smaller number of templates). The software also matches small segments of the signal (typically 0.025 seconds in duration). This enables it to "catch" a call from within a noisy dawn chorus in a fortuitous time slot that a human might miss. Humans

need at least 0.100 seconds to process a sound and quite a bit longer to make any real sense of it.

The reference library that the software uses is almost unlimited (it could have more than a billion separate calls in its reference library), which far exceeds the capacity of most humans.

In tests we have run against acknowledge avarian experts the software excels in both accuracy and speed.

3.6 Real-time Recognition

We are currently building a real-time version of this system (called an Autonomous Recording Unit (ARU)). The system is a PC based recorder that will process what it records in real-time and have I/O (input-output capabilities). Uses for this include activating an SMS or other device when a particular species is head (e.g. a rare bird, frog or bat), issuing alarms to deter the recognized species from entering the area (e.g. birds on runways) and categorizing sounds in vocalization studies (e.g. real-time vocalization recognition of dolphin calls).

5 Conclusion

We have demonstrated that it is possible to exceed the capabilities of even a human expert with the software that we describe so long as the settings for the templates are appropriate. Entirely new possibilities including large scale acoustic surveys, acoustics searches for rare and endangered species are now feasible.

References

[1] Jinnai, M. Boucher, N J. Robertson , J. Kleindofer, S. “Design considerations in an automatic classification system for bird vocalization using the two dimensional Geometric Distance and Cluster Analysis”, International Congress of Acoustics, Sydney, August (2010)