Paper presented at the 1995 IEEE Systems, Man and Cybernetics Conference, October 22-25, 1995, Vancouver, Canada.

Pattern Playback from 1950 to 1995

Malcolm Slaney
Interval Research, Inc., 1801-C Page Mill Road, Palo Alto, CA 94304
malcolm@interval.com

ABSTRACT

This paper describes algorithms to convert spectrograms, cochleagrams and correlograms back into sounds. Each of these representations converts sound waves into pictures or movies. Techniques for inversion, known as the Pattern Playback problem, are important because they allow these representations to be used for analysis and transformations of sound. The algorithms described here use convex projections and intelligent phases guesses to iteratively find the closest waveform consistent with the known information. Reconstructions from the spectrogram and cochleagram are indistinguishable from the original sound. In informal listening tests, the correlogram reconstructions are nearly identical.

1. INTRODUCTION

Audio researchers often use visual representations of sound to gain a better understanding of the components of the sound. This paper describes methods for displaying sound visually and explains algorithms developed to turn these pictures back into sound. We call these inversion techniques Pattern Playback. Figure 1 shows the techniques described in this paper.

Although there are many ways to represent sounds, this paper only describes those that are used to make images or movies. These representations include the conventional spectrogram and two analytic models of human perception. Other sound analysis techniques, such as linear-predictive coding (LPC) and sinusoidal analysis are not covered here. These techniques only model a subset of the sounds we hear and generally aren't used to make visual representations.

Pattern playback is interesting for two reasons. First, pictures of sound are useful for describing and transforming sounds. Editing a waveform is a good solution for some tasks, but not for all sound transformations. For some tasks editing a spectrogram might be more appropriate. Thus, using the techniques described in this paper, the most appropriate representation for the task can be used and the desired sound resynthesized from its picture. Moreover, pattern playback is a good test of the quality and completeness of a representation.

An ideal representation will have several properties. First it will make clear the salient patterns of a sound. Second, small changes in the representation make small changes in the resulting sound. Finally, each picture or movie will represent a perceptually unique sound. The three visual representations described here meet these requirements to varying degrees.

The name Pattern Playback was used by Frank Cooper in the early 1950's. Cooper showed that it was possible to draw a pattern of paint splotches on plastic and then use a machine of his design to play back the sound [1]. This made it possible for his lab to do many psychoacoustic experiments and it helped validate the use of a spectrogram. Today, the analysis and resynthesis tools are more powerful. This paper will describe Cooper's original machine as well as the state of the art.

This paper has five additional sections. Section 2 describes the mathematical tools used by the inversion techniques described here. Sections 3, 4, and 5 describe the use and inversion of the spectrogram, cochleagram and correlogram. Each of these representations will be described in its respective section. The cochleagram and correlogram are computer models of auditory perception that have many interesting properties. Only in the last year have we discovered techniques to invert them. Finally, Section 6 will review our successes. The techniques described in this paper have also been discussed in other publications [2, 3]. New results are presented here.

2. TOOLS

The algorithms used in this paper have much in common. In the analysis case the representations are generated by transforming the sound's waveform into a higher-dimensional space. The forward transform is well defined, stable and results in a picture or movie of the sound.

In general the inverse process is not so straightforward. To address this problem we define the inverse task as finding a waveform that comes closest to generating the given picture. This last statement has two components. First, many waveforms are often possible, we just need to pick one. If the representation is perceptually relevant, then all possible waveforms sound identical, even though the waveforms are very different. Second, not all pictures are valid sounds. We should find a waveform that produces a picture as close as possible to the original picture. Choosing the closest answer is an important aspect of the algorithms.

This section describes the two primary tools we used for pattern playback: convex projections and reducing the RMS error.

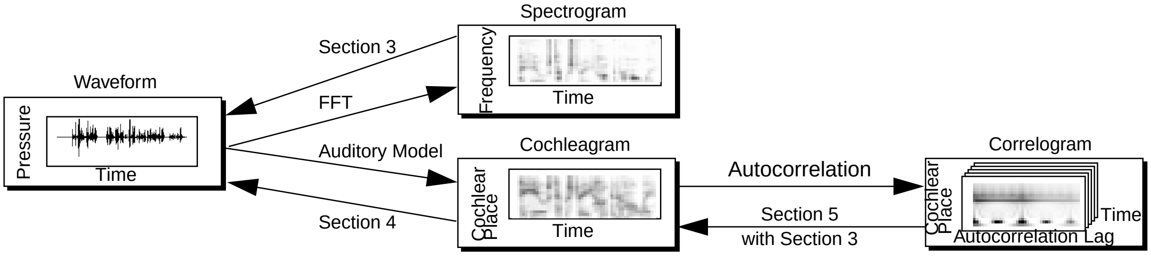

Figure 1 - This figure shows a roadmap of this paper, the auditory representations and the pattern playback techniques.

Convex Projections

Convex projections form the basis of the inversion techniques described in this paper. In each case we can describe the desired solution as the intersection of a number of infinite dimensional sets. In some cases the set is as simple as all functions that have a finite bandwidth. Other constraints are more specific: all functions, f(x), that have the value 3.4 when x=1.

We solve the inversion problem by finding a waveform that satisfies the known constraints on the solution [4]. Ideally all constraints will overlap at a single point. In practice there is a large set of waveforms that satisfy the constraints and we need to pick one, perhaps the waveform that is closest to our initial guess. We hope that the union of all constraints is a small neighborhood of perceptually identical waveforms. In this case it doesn't matter which waveform we choose.

The term *convex projections* indicates two components of the process. First, a constraint must restrict the solution to a convex set. A set of points is convex if given any two points in the set, all intermediate points are also in the set. Thus convex sets are important because they allow us to define a projection operation. If the set is convex, given any point outside the set there is a unique point inside the set that is closest to the original point. The process of moving a point which is outside the set to the closest point in the set is called a projection.

The algorithm we use to invert each of the sound representations consists of a number of convex projections. We can start with an initial guess and at each step we project the waveform onto one of the convex sets. At the next step we project onto another set, and iterate between sets until we have converged to a solution.

While in theory we can start the iterations with any waveform, in practice we want to start with the best possible guess. Convergence of these algorithms depends on the quality of the initial estimates and the constraints used to define the convex projections. Depending on the orthogonality of the constraints and how far we need to move, convergence might be too slow to be practical. If we generate a better initial guess, then the convex projections are not nearly so critical. In many cases a good initial guess dramatically reduces the number of projections.[1]

Convex sets can be defined in many different ways. In some cases, the set will be defined in the time domain; in other cases, it might be defined in the frequency domain. There are many types of constraints that we and the given picture can place on the final result. The list below shows the constraints used in this work and the corresponding projection operator:

- Bandlimited - Filter to remove unwanted components
- Subset of known time-domain values - Replace estimated values with known values.
- Unique magnitude - Not convex, but keep phase and replace with known magnitude.

RMS Energy

The error metrics in these inversion techniques are based on reducing the RMS energy in the difference signal. We would prefer to optimize the perceptual error, but only RMS error is mathematically tractable. As we will show, our results do not suffer due to this limitation in our algorithms. Still there is no guarantee that the methods we describe will necessarily sound perfect.

Through all these procedures, we're only reducing the RMS difference between the given picture and the picture of the current waveform guess. There are many modifications of a spectrogram that are imperceptible. Adding a tone that is below the auditory masking threshold will have a large effect on the spectrogram. Reducing the RMS level of this component in the spectrogram will have no effect on the perception.

We hope the perceptual auditory representations are less sensitive to this effect. As we described above, a perceptual representation should model sounds so that small changes in the perception correspond to small changes in the representation. In the right perceptual space, RMS differences will accurately model perception.

3. SPECTROGRAMS

Usage

Spectrograms are a popular way to visualize sound. Spectrograms, or more correctly short-time Fourier transforms (STFT), transform small portions of a waveform into the frequency domain. Transforms from adjacent windows of data are rendered as a picture to create an image of the sound's frequency content versus time.

Used originally to analyze speech, the spectrogram took on new meaning when it was used to synthesize sounds. Cooper's machine used an array of light sources, each modulated at one of the fifty harmonics of 120Hz, to illuminate acetate tape. Patterns were painted on the film and the light that was reflected from the pattern was collected and amplified for playback. The result was "highly intelligible" speech [1].

Inversion

In principle, since the Fourier transform is easily inverted, computers should be able to replicate the original pattern playback hardware. But a spectrogram usually starts with overlapping windows of data, and the phase of each Fourier transform is thrown away. The result is a nice picture of the spectral content, but not something that is easy to invert. Not only is it necessary to recover the lost phase, but it's important to combine the multiple windows of data to come up with the best reconstruction.

Spectrogram inversion is accomplished by noting that there are two defining sets. First the desired waveform has a known magnitude spectrum. We also know that all windows of data must be consistent. Neither of these restrictions are convex sets, but it is easy to show [5] that the following procedure always reduces the error at each iteration. The result is an algorithm that should converge to the correct answer.

Spectrogram inversion is accomplished by a three step process: initial phase estimation, time-domain projection, and frequency-domain projection. The primary goal is to recover the lost phase information. We are given the exact magnitude information. If we find the exact phase information, we can generate the original waveform.

To reduce the need for iterations, we should start with a guess for the phase that will lead to a good reconstruction. One possibility is to generate the waveform, from left to right, inverting each slice of spectral data assuming zero phase and merging it into the waveform that we've already computed.

[1] Thanks to Richard F. Lyon for emphasizing this aspect of our work.

We can generate a better initial waveform by estimating a consistent set of phases. This is found by rotating each new window of data over the existing waveform looking for the best match [2]. The peak of the cross-correlation between the existing waveform and the new data slice indicates the proper rotation and the best set of phases for the reconstruction.

The second stage of the spectrogram inversion is time-domain projection, combining multiple windows of time-domain data. The basic procedure is called overlap-and-add. This work uses a variation derived by Griffin [5] that minimizes the mean-squared error in the reconstruction. The result, at this point, is the first waveform estimate.

The third stage is frequency-domain projection, combining the STFT of the estimated waveform and the magnitude of the original spectrogram. We know that the original spectrogram has the correct magnitude thus we take the magnitude from the original spectrogram and the phase from the new spectrogram. The algorithm repeats the last two stages as often as needed.

The properties of this algorithm are easiest to see by reconstructing the spectrogram of a single tone. Figure 2 shows the reconstruction assuming zero phase (no iterations) and after five iterations. Note the large phase discontinuities in the initial waveform (a) make it difficult for the algorithm to converge. Figure 2(b and c) show the result if the rotation algorithm described above is first used. Now the initial waveform (c) is as good as the zero-phase iterated case (b) and the extra iterations have little work to do (d).

It is worth noting that using autocorrelation to align the phase of the initial STFT produces results identical to the SOLA algorithm [6] used to perform time-scale modification of speech. The extra iterations serve to improve the resulting waveform and thus might be useful in other applications where modifying the time course or pitch of a signal is required.

Results

It is important to remember that not all spectrograms correspond to valid waveforms. Figure 3 is a reproduction of a portion of one of Cooper's painted spectrograms. We can scale this so that formant frequencies are in the proper place and generate a reconstruction. Spectrograms with uniform black patches do not correspond to valid waveforms. The inversion result is buzzy and the spectrogram of this inversion has a large amount of speckle noise. The closest the spectrogram can come to representing the uniform black spectrogram patches is shown in Figure 3b.

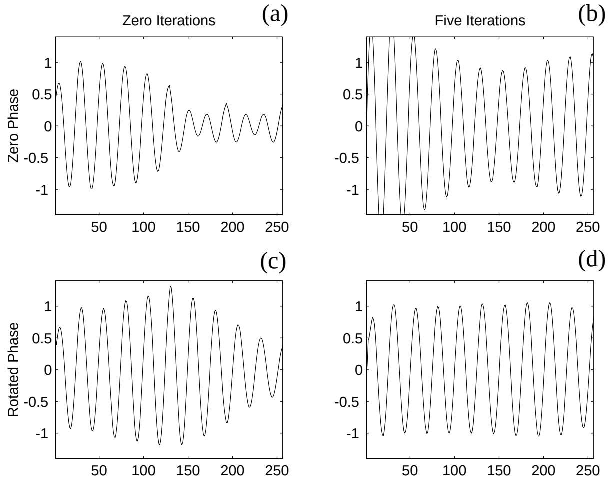

Figure 2 - Four spectrogram reconstructions of a constant amplitude tone showing the effect of initial phase estimates and iterations. The first iteration with rotated phase (c) is as good as the final iterated guess with zero phase (b).

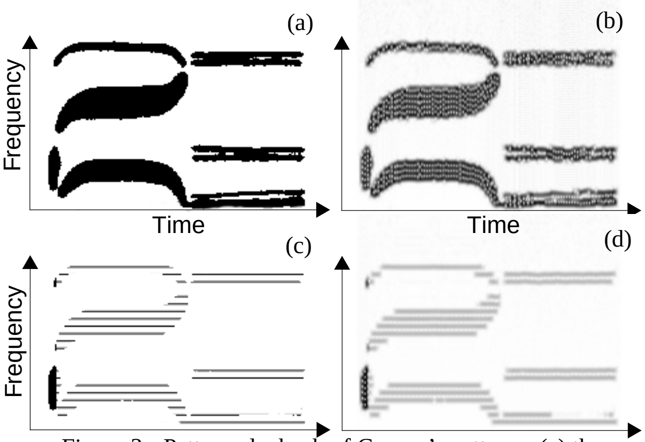

Figure 3 - Pattern playback of Cooper's patterns: (a) the original spectrogram pattern, (b) spectrogram of inverted signal, (c) original spectrogram with pitch harmonics, (d) spectrogram of inverted signal with pitch.

Unlike the Pattern Playback machine, the spectrogram in Figure 3a is missing any pitch information. Figure 3c shows what happens when we multiply the original spectrogram by the spectrogram of a 120 harmonic series. This simulates the glottal source of a monotone speaker. The inversion process produces a speech sound with a clear pitch. In addition, the spectrogram of the inversion result is very close to the original filtered spectrogram (Figure 3d.)

The remainder of this paper discusses the use and inversion of perceptual audio representations. We will return to the spectrogram shortly since the correlogram inversion depends on spectrogram inversion.

4. THE COCHLEAGRAM

Uses

Psychoacousticians have built many models of human auditory processing. Some of these models are detailed, while other only vaguely resemble the auditory system. They all share the desire to capture the most important aspects of the way that sound is perceived by humans. The output of these models is a physical measure at some stage in the auditory processing stream.

All cochlear models share the goal of modelling the time-frequency analysis properties of the cochlea. Thus their output is a measure of cochlear activity as a function of place along the cochlea. Since each position along the cochlea responds best to one frequency, we often think of cochlear models as representing the signal as a function of time and frequency.[2] The result is a two dimensional representation of sound.

[2] This is not strictly true since the cochlea is highly nonlinear [7].

In this work, we use a cochlear model designed by Richard F. Lyon [8]. Sound is filtered by a cascade of band-pass filters. The output of each filter is half-wave rectified to simulate the hair cell detector and then passed through a coupled multi-stage automatic gain control (AGC) to simulate the adaptation processes. Each filter output, after detection and AGC, is called a channel; typically 50-100 channels are used to model the cochlear processing. The output of this model is a sample-by-sample estimate of the probability of firing of the auditory nerve fibers.

Inversion

The inversion process for this cochlear model includes three stages

- Invert the AGC,
- Replace portions of the waveform lost by the detector, and
- Sum channels correctly.

We will describe the inversion process from output back to input. A similar process is also possible with other cochlear models designed by Irino [9] and Shamma [10].

Each stage of the AGC uses its output level to set its gain. Since the AGC gain is directly observable, we can easily compute the state of each AGC stage at any point in time. Each AGC stage is inverted by dividing the known AGC output, by the calculated gain. There are issues of stability, but these are minimized by keeping the input signal level low. High input signals lead to very small AGC gains. When the AGC is inverted by dividing these small numbers into the signal, noise and numerical imprecision lead to errors. This is an issue when using the output of the correlogram inversion process described in the next section.

Inverting the (half-wave) detector non-linearity is a classic use of convex projection. For each channel of cochlear output, we know the positive values of the filter output. We also know that each filter output has a relatively narrow bandwidth. We can use these two constraints to invert the half-wave rectifier. Bandpass filtering works especially well since one effect of rectification is to add harmonics of the original signal.

An even more efficient algorithm is possible since we know each channel is related to its neighbors. In the end, we want to "invert" each cochlear filter and combine channels into a single waveform. Thus we can postpone the half-wave rectifier inversion by combining it with the filter inversion.

The final step, inverting a bank of filters, is easiest if we realize that we only care about the ensemble of filters. It is simple to invert the transfer function of a filter, thus adding gain to the signal where it was first attenuated. But this is not a good solution since each filter removes large portions of the spectrum. A better solution is to use the information from other channels.

We can combine channels for the filterbank inversion by correcting for the filter's phase characteristics and then summing all the phase-corrected channels. We might, for example, have 60% of the energy at any one frequency in channel 42, and another 30% in channel 43. If we invert the filter-induced phase of these two channels we only need to increase the gain a bit to restore the amplitude of the signal at that frequency.

Phase correction is accomplished by either running the signal backwards through the original filter, or reversing the filter's impulse response so it is noncausal.[3] The effect is the same in either case. The spectral gain seen by a channel in the filter bank is squared. The effect of this is minimal since the gain in each filter's passband is close to one. More importantly, the phase change at each frequency is cancelled out. After inverting and summing all channels we are left with a waveform that has exactly the right phase, but an overall spectral tilt due to the spacing of the filters. This tilt is gentle compared to the original bandpass filter response and thus easy to correct.

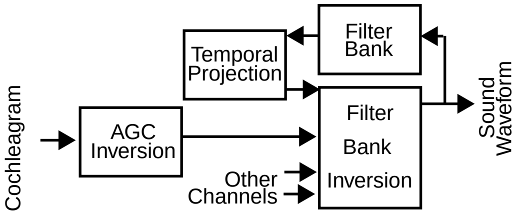

Figure 4 - A more efficient method for inverting the detector and the filter bank is shown here.

We now return to the subject of the half-wave rectifier inversion. We know that each channel is bandlimited. We could design a special filter to do the projection, but a more efficient solution is to combine this stage with the filtering performed for the filter-bank inversion. The more efficient scheme is diagrammed in Figure 4.

Results

In general, results of the cochlear inversion are perceptually identical to the original waveform. As long as numerical instabilities have not overwhelmed the AGC inversion, the procedure is well behaved. The half-wave rectifier inversion is accurate. What little information is not available from the original waveform is easy to recover because there are two to four channels covering every frequency. The negative portions of a waveform not present in one channel will often be at a slightly different phase in the adjacent channel and easy to incorporate.

Figure 5 shows typical reconstructions from cochleagrams.[4] The top plots show reconstructions of an impulse. Extra iterations do improve the result, removing noise from the initial reconstructions.

Like the spectrogram, the cochleagram inversion can be used to generate interesting sound transformations. More importantly it also gives us a way to simulate the effects of hearing deficiencies. One such deficiency is the loss of compression that often accompanies old age. The effect of such a simulation is shown in Figure 5. The bottom-left quarter of this figure shows a normal inversion. The right half shows the result without inverting the AGC. The result is a sound that is highly compressed, much the way the auditory system compresses the signal.

[3] This is exactly the procedure used in analysis/resynthesis filter banks.

[4] The syllable "tap" used in many examples in this paper are samples 14000 through 17000 of the "train/dr5/fcdf1/sx106/sx106.adc" utterance in the TIMIT Speech Database.

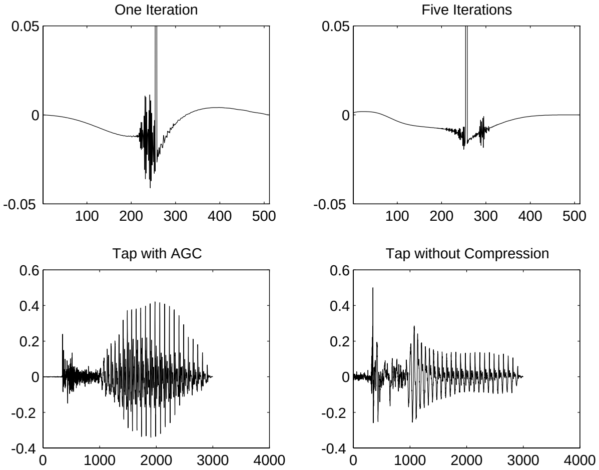

Figure 5 - The plots on top are magnified cochleagram reconstructions of an impulse. The gentle curve is caused by the loss of low-frequency information in the cochlear model. The bottom plots show reconstructions of the word "tap" with and without the AGC inversion.

5. CORRELOGRAMS

Use and Calculation

Both spectrogram and cochleagrams are complete representations of a sound. That is, they both encode all the information in the sound. But neither encoding method is perceptually useful.

The correlogram has been proposed as the next stage of processing after the cochlea. Auditory nerve firings are analyzed in such a way as to summarize the periodicities in the neural firing rates. These periodicities are key to understanding pitch perception and we believe are the fundamental representation that makes it possible for us to understand one of many simultaneous speakers [7].

Unlike the previous representations, a correlogram is a three-dimensional representation of sound. Figure 1 shows the sequence of steps that lead to a correlogram and one frame of a correlogram movie. Each frame of the correlogram shows frequency (or more precisely cochlear position) along the vertical axis. The horizontal axis is a function of autocorrelation delay, and the movie frames change with time.

Correlograms are interesting because the two most important components of a sound, the pitch and the overall spectral shape, are displayed on orthogonal axes. The pitch shows up as a dark vertical line at the autocorrelation time delay corresponding to the dominant periodicity in the audio signal. Formants, and other characteristics of the audio spectrum, show up as horizontal bands.

Correlograms are calculated using autocorrelations of each cochlear channel's output. At each time we wish to sample the output movie, we compute an autocorrelation of each channel's recent neural firing rate. The neural data is windowed, FFTs are calculated, squared, and then inverted to arrive at the autocorrelation. By assembling all of these autocorrelations in the proper order we can build a correlogram movie.

Inverting the correlogram takes two steps. First we need to invert the autocorrelation calculation to arrive at an estimate of each channel's cochlear output. Then the cochlear output can be inverted to arrive at an estimate of the original waveform. The next two subsections of this paper talk about the autocorrelation inversion, and steps that can be taken to speed up the convergence.

Inversion–Transform to Spectrogram

Correlograms are easy to invert by noting that autocorrelation is related to the power spectrum of a signal. For each cochlear channel, we assemble all the autocorrelations and calculate the corresponding power spectrums. The power spectrum, for each window of data, is equal to the square of the spectrogram magnitude and can be inverted as described in Section 3 of this paper.

This process is shown in Figure 6. The resulting spectrogram is interesting because it represents a narrow band waveform that has been half-wave rectified. The half-wave rectification adds the harmonics that are seen.

Inversion–Phase Guesses

Like all convex projection procedures, better initial guesses dramatically reduce the number of iterations needed to achieve convergence. In correlogram inversion we can propagate the phase in two different directions. As described in spectrogram inversion, we can predict the phase as we build the waveform from left to right using

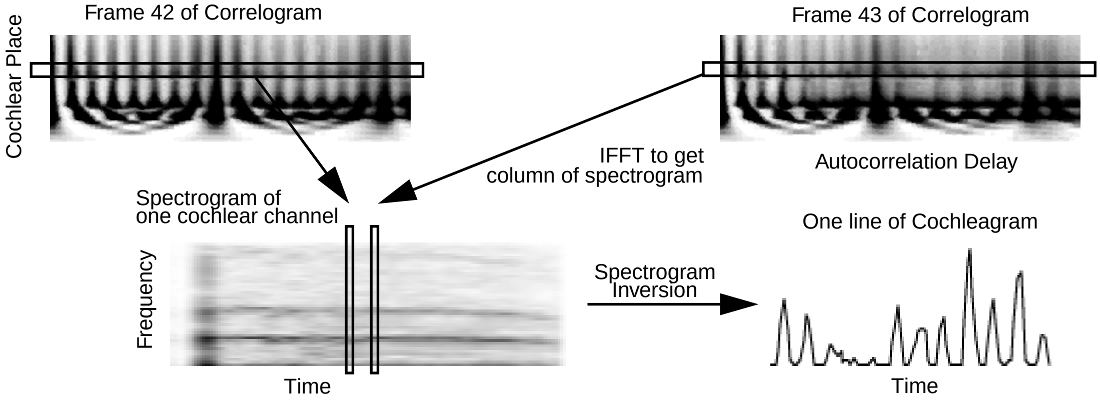

Figure 6 - The steps in correlogram inversion are shown above. Each channel of a correlogram is converted to a power spectrum using an inverse FFT. Then the spectrogram can be inverted to find the output of the cochlear channel. The dark vertical line in the spectrogram is from the "t" in "tap". The horizontal bands show the passband and the harmonic distortion due to the half-wave detector. (Adapted from [3])

cross-correlation. Similarly, in correlogram inversion we can use the final phase estimate from one channel to initialize the phase for the next channel. This works since one channel overlaps spectrally with its neighbor.

Results

Reconstructions from correlograms are shown in Figure 7. A series of impulses and the word “tap” are used as input to a cochlear model and a correlogram frame is calculated every 64 samples. Each channel of the correlogram is inverted as shown in Figure 6 to estimate the waveforms shown.

On the left of Figure 7, the reconstructions are done without any iterations and the resulting sounds are muffled. The pitch is correct because the initial channels are lined up using the phase-rotation method described in Section 3. The images on the right of Figure 7 are iterated reconstructions. Ten iterations of the spectrogram inversion are used for the first channel (high-frequency channel), and three iterations are used for each successive channel. Lastly, ten iterations of the cochleagram inversion are used to estimate the final sound waveform. The iterated reconstructions sound nearly identical to the original waveforms.

The impulses do not line up as well as the original waveform due to phase mismatches across the filterbank. Unless the entire correlogram inversion process is iterated, thus closing the loop, the relative timing between the first and the last channel of the correlogram inversion will not necessarily be synchronized. However, this relatively modest phase change across the entire range of hearing does not affect our perception.

6. CONCLUSIONS

We have demonstrated pattern playback from three different types of auditory representations. Reconstructions from the spectrogram and cochleagram are indistinguishable from the original sound. In informal listening tests, the correlogram reconstructions are nearly identical. This is noteworthy since compared to other representations more sounds can be represented (than LPC) and there are fewer thresholds to set (than sinusoidal analysis).

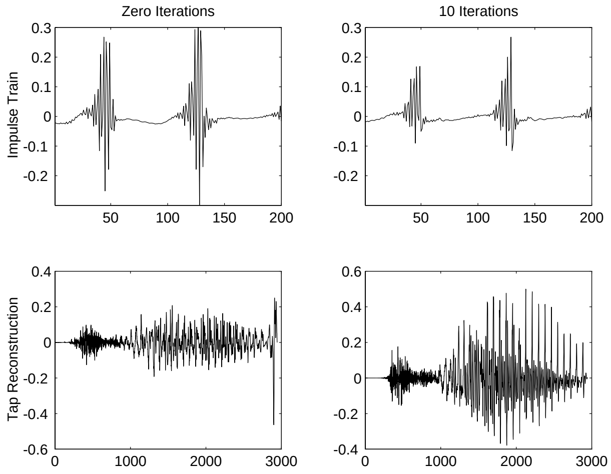

Figure 7 - Four correlogram reconstructions are shown, with and without iterations. The top two are from an impulse train, the bottom two are of the word “tap”. The reconstructions without iterations are muffled while the iterated results sound nearly identical to the original waveform.

Convex projections are the key to these techniques, but better initial estimates make a big difference in the quality of the reconstruction. A new algorithm described here for rotating the phase of the spectrogram inversion is perhaps the most important key to the successes. Future work will look at reconstructions from partial information.

7. ACKNOWLEDGEMENTS

I am grateful for the assistance, both technical and editorial from Michele Covell. Richard F. Lyon and Daniel Naar helped develop the original algorithms described here. Many conversations with Richard O. Duda, Shihab Shamma and the participants of the Stanford CCRMA Hearing Seminar shaped this work in many important ways. Lisa Stifelman helped with the editing of this paper. Finally, I appreciate the encouragement and support I have received from J. Marty Tennenbaum, Frank Cooper and Meg Withgott.

8. REFERENCES

[1] F. S. Cooper, “Some Instrumental Aids to Research on Speech,” *Report on the Fourth Annual Round Table Meeting on Linguistics and Language Teaching*, Georgetown University Press, pp. 46-53, 1953.

[2] Malcolm Slaney, D. Naar, R. F. Lyon, “Auditory model inversion for sound separation,” *Proc. of IEEE ICASSP*, Volume II, pp. 77-80, 1994.

[3] Malcolm Slaney, “Pattern Playback in the 90’s,” in *Advances in Neural Information Processing Systems 7*, Gerald Tesauro, David Touretzky, and Todd Leen (eds.), MIT Press, Cambridge, MA, 1995.

[4] R. W. Papoulis, “A new algorithm in spectral analysis and band-limited extrapolation,” *IEEE Trans. Circuits Sys.*, vol. 22, p. 735, 1975.

[5] D. Griffin and J. Lim, “Signal estimation from modified short-time Fourier transform,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 32, 236-242, 1984.

[6] E. Moulines and J. Laroche, “Non-parametric techniques for pitch-scale and time-scale modification of speech,” *Speech Communications*, vol. 16(2), pp. 175-205, 1995.

[7] Malcolm Slaney and R. F. Lyon, “On the importance of time—A temporal representation of sound,” in *Visual Representations of Speech Signals*, eds. M. Cooke, S. Beet, and M. Crawford, J. Wiley and Sons, Sussex, England, 1993.

[8] R. F. Lyon, “A computational model of filtering, detection, and compression in the cochlea,” *Proc. of the IEEE ICASSP*, pp. 1282-1285, 1982.

[9] T. Irino, H. Kawahara, “Signal reconstruction from modified auditory wavelet transform,” *IEEE Trans. on Signal Processing*, vol. 41, pp. 3549-3554, Dec. 1993.

[10] X. Yang, K. Wang, and S. Shamma, “Auditory representations of acoustic signals,” *IEEE Trans. on Information Theory,* vol. 38, pp. 824-839, 1992.