Jonghyun Lee, Roger B. Dannenberg, and Joohwan Chun. 2004. "Cancellation of Unwanted Audio to Support Interactive Computer Music." In *The ICMC 2004 Proceedings*. San Francisco: The International Computer Music Association, pp. 692-698.

Cancellation of Unwanted Audio to Support Interactive Computer Music

Jonghyun Lee[†], Roger B. Dannenberg[‡], and Joohwan Chun[†]
[†]Scientific Computing Laboratory, Department of Electrical Engineering
Korea Advanced Institute of Science and Technology,
373-1 Kusong-Dong, Yusong-Gu, Taejeon, 305-701, South Korea
jhlee@sclab.kaist.ac.kr, chun@sclab.kaist.ac.kr
[‡]School of Computer Science, Carnegie Mellon University
dannenberg@cs.cmu.edu

Abstract

A real-time unwanted-audio cancellation system is developed. The system enhances recorded sound by canceling unwanted loudspeaker sounds picked up during the recording. After cancellation, the resulting sound gives an improved estimation of the live performer's sound. The cancellation works by estimating the unwanted audio signal and subtracting it from the recorded signal. The canceller is composed of a delay block and two adaptive digital filters. Our work extends conventional echo-cancellation methods to address problems we encountered in music applications. We describe a real-time implementation in Aura and present experimental results in which the proposed canceller enhances the performance of a real-time pitch detector. The cancellation ratio is measured and limitations of the system are discussed.

1 Introduction

In interactive computer music performances, a live musician often uses a microphone to capture an acoustic performance, but the microphone also picks up sounds from the computer. To enhance the quality of the recorded sound and to reduce the capture of other sounds, we usually use a good directional microphone and locate it as close as possible to the player, but some computer sound is still captured. This unwanted sound can interfere with the signal analysis and signal processing typically used by interactive music systems.

Signal processing systems can use various methods to estimate unwanted signals and subtract them from the recorded sound. One approach synthesizes an inverted signal to cancel the unwanted signal acoustically (Kuo and Morgan 1999).

In this paper, we describe an application of cancellation to enhance recording quality. We have created a real-time implementation of the system using moderate computational power. An important difference between this and previous work is that our goal is to enhance interactive music performance systems. The often-independent nature of computer-generated "unwanted" sounds and the sparseness of musical spectra can make it very difficult to estimate the system characteristics. This has led us to develop new techniques, which we describe below.

For this work, we assume interactive music performance systems with a single microphone and a single loudspeaker, although extension to more channels should be possible. The microphone captures the sound of one or more instruments. Simultaneously, sounds generated by a computer are played through a loudspeaker. In these situations, the microphone often picks up unwanted sound from the loudspeaker. This unwanted sound may degrade the computer analysis of the acoustic instrument.

We want to enhance the recorded signal quality by canceling the unwanted sound. Note that the computer-generated sound is available in digital form, so the real problem is to estimate how this sound is transformed on the path to the loudspeaker, through the room, to the microphone, and back to the computer. If we can estimate this entire channel, then we can digitally simulate the effects of the channel on the computer-generated signal and subtract the result from the signal obtained from the microphone.

At first glance, one might guess that the computer-generated signal can simply be delayed, attenuated, and subtracted from the microphone signal to accomplish our goal. Unfortunately, the frequency-dependent behavior of the loudspeaker and microphone have a large effect on the signal. Thus, the system must estimate the overall response of the entire signal path, or channel, to achieve cancellation. To make matters even worse, the channel is not fixed. As the performer moves, the

Jonghyun Lee, Roger B. Dannenberg, and Joohwan Chun. 2004. "Cancellation of Unwanted Audio to Support Interactive Computer Music." In *The ICMC 2004 Proceedings*. San Francisco: The International Computer Music Association, pp. 692-698.

channel characteristics change. For example, the performer might move into the direct path from loudspeaker to microphone, or the performer might accidentally move the microphone.

Our cancellation system is implemented as a signal processing component of Aura (Dannenberg and Brandt 1996). Using pre-recorded sound files to simulate both performers and computer-generated signals, we can experiment with different configurations and listen to the results. One important application of this work is to isolate the wanted audio signal for analysis. We experimented with the cancellation system as a front-end to a pitch estimation module and show that pitch estimation is enhanced by canceling unwanted sounds.

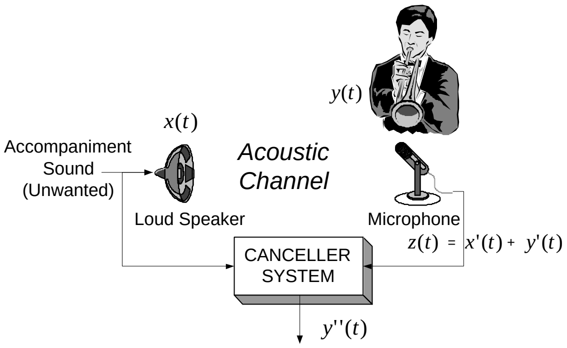

Figure 1: The overview of sound canceller system.

2 Related Work

These are not entirely new problems, and adaptive systems already exist for active noise cancellation (Kuo and Morgan 1999) and echo cancellation (Haykin 1969). However, in an active noise canceller, the noise is continuous and spectrally stable, and in an echo canceller, the echo signal is fairly deterministic. For example, in telephony, the echo comes from relatively stable electrical circuits rather than changing acoustic environments. In telephony, we believe the channel varies more slowly than in a live music performance. Also, telephony uses a training signal to analyze the circuit before talking begins.

In our application, the wanted signal can interfere with our estimation of the unwanted signal, but we do not know when the wanted signal starts or stops. Also, when the unwanted signal is absent or weak, it is impossible to estimate the channel or to estimate the canceling signal. Therefore, we develop a method to evaluate when the adaptation leads to improvement. When there is no improvement, the channel estimate is not updated.

In the field of telephony, this is called the "double-talk problem." (J. Benesty and Cho 2000; Kuo and Pan 1993) Usually, only one speaker talks at a time, and the double-talk situation is often detected by comparing the incoming signal with echo to a threshold. Adaptation is stopped when double talk is detected. With live music, the "double-talk" situation is the norm rather than the exception.

3 Cancellation system

Figure 1 offers an overview of our cancellation system. $x(t)$ is an accompaniment signal played from the loudspeaker. A musician is playing near a microphone producing the waveform $y(t)$. The sound of the accompaniment and target instrument are recorded together, so the recorded waveform is the sum of two sounds. Because of the effect of the acoustic channel, the recorded sound is not exactly identical to $x(t)$ and $y(t)$, so we define the recorded sound as $z(t) = x'(t) + y'(t)$, where $x'(t)$ and $y'(t)$ are distorted versions of $x(t)$ and $y(t)$, respectively. We want to get only the target sound by canceling the sound of the loudspeaker. The cancellation system inputs are the sound sent to the loudspeaker and the sound received from the microphone. The output is the cancelled sound which is written as $y''(t)$.

The recorded sound $x'(t)$ is not a simple time-delayed copy of $x(t)$ because of many effects including sound reflection and diffraction in the acoustic environment. Other effects are the transfer characteristic of the loudspeaker, microphone, amplifier and recording equipment, including quantization errors in the A/D and D/A converters, the nonlinearity of the amplifier, and frequency characteristic of the loudspeaker and microphone. All these effects are referred to collectively as the acoustic channel. When the player or microphone moves, the channel varies. Therefore the recorded sound, which is now a discrete signal indexed by n, is given mathematically as

$$\begin{aligned} z(n) &= x'(n) + y'(n) \quad (1) \\ &= h(n) * x(n) + y'(n) + n_g(n) + n_q(n) \\ &\quad + n_n(n) \quad (2) \end{aligned}$$

where $n_g(n)$ is background noise, and we assume its probability density function (PDF) is white Gaussian. $n_q(n)$ is quantization noise due to A/D conversion, and $n_n(n)$ is the sum of unknown noises due to nonlinearities. n is the time index and $*$ is the convolution operator. The PDF of $n_q(n)$ is given as

$$P(n_q) = \begin{cases} 1/Q & -Q/2 \le n_q \le Q/2 \\ 0 & \text{elsewhere} \end{cases} \quad (3)$$

where Q is number of quantization steps determined by the number of bits w: $Q = 2^{-(w-1)}$.

If we know the channel $h(n)$ exactly, we can estimate the sound $x'(n)$. When the target sound $y'(n)$ does not exist, the problem is similar to system identification problem shown in Figure 2.

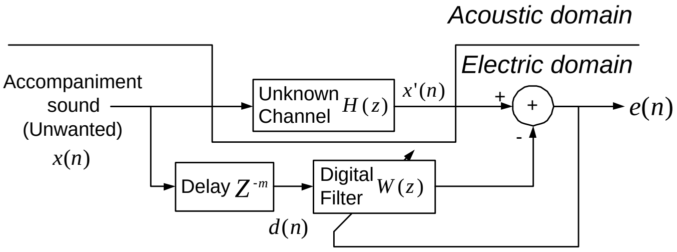

Figure 2: System identification block diagram.

The delay block in Figure 2 compensates for the delay of the D/A system, the acoustic delay, and the capture delay of the A/D system. If the digital filter $W(z)$ is long enough, the delay block is not required, but a long filter is undesirable because it requires additional computation and memory. We use the Phase Transform (PHAT) delay estimation to estimate time delay (Knapp and Carter 1976; Ianniello 1982; Carter 1987). PHAT require two real fast Fourier transforms (FFT) and one complex FFT transform.

$$\tau_{PHAT} = \arg\max R_{x'd}(\tau) \tag{4}$$

$$\begin{aligned} R_{x'd}(\tau) &= \int_{-\infty}^{-\infty} \frac{X'(\omega)D(\omega)}{|X'(\omega)D(\omega)|} e^{j\omega\tau} d\omega && (5) \\ &= FFT\left\{ \frac{X'(\omega)D(\omega)}{|X'(\omega)D(\omega)|} \right\} && (6) \end{aligned}$$

where $X'(\omega)$ and $D(\omega)$ is the real FFT of $x'(n)$ and $d(n)$ respectively, and $d(n)$ is the delayed version of $x(n)$.

After estimating delay, we must estimate the digital filter. The objective of the adaptive filter is to minimize a residual error signal $e(n)$. We want $e(n) = 0$ after the adaptive filter $W(z)$ converges. The digital filter $W(z)$ can be estimated using the LMS or RLS algorithm (Haykin 1969). In our system, we use the LMS adaptation algorithm because the computation is feasible in real time. The residual signal is expressed as

$$e(n) = x'(n) - \mathbf{w}^T(n)\mathbf{d}(n) \tag{7}$$

where n is the time index, $\mathbf{w}(n) = [w_1(n)\, w_2(n) \cdots w_L(n)]^T$ and $\mathbf{d}(n) = [d(n)\, d(n-1) \cdots d(n-L+1)]^T$ are the coefficient and signal vectors, respectively, ^{T} is the transpose operation and L is the filter order. The filter $W(z)$ must be of sufficient order to accurately model the impulse response of the acoustic channel.

Assuming a mean square cost function $\xi(n) = E[e^2(n)]$, the adaptive filter minimizes the instantaneous square error $\hat{\xi}(n) = e^2(n)$. Using the steepest descent algorithm, we update the coefficient vector in the direction of the negative gradient with the step size μ:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2}\nabla\hat{\xi}(n) \tag{8}$$

where $\nabla\hat{\xi}(n)$ is an instantaneous estimate of the mean squared error (MSE) gradient at time n and is expressed as

$$\begin{aligned} \nabla\hat{\xi}(n) &= \nabla e^2(n) = 2[\nabla e(n)]e(n) && (9) \\ &= -\mathbf{d}(n)e(n) && (10) \end{aligned}$$

Therefore we have the LMS adaptation

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{d}(n)e(n) \tag{11}$$

The performance of the cancellation system can be determined by a frequency-domain analysis of the residual error signal $e(n)$. The autopower spectrum of $e(n)$ is: (Kuo and Morgan 1999)

$$S_{ee}(\omega) = [1 - C_{dx'}(\omega)]S_{x'x'}(\omega) \tag{12}$$

where $C_{dx'}(\omega)$ is the magnitude-squared coherence function between $d(n)$ and $x'(n)$, and $S_{x'x'}(\omega)$ is the auto power spectrum of $x'(n)$. The magnitude-squared coherence function is defined as

$$C_{dx'}(\omega) = \frac{|S_{dx'}(\omega)|^2}{S_{dd}(\omega)S_{x'x'}(\omega)} \tag{13}$$

and if $x'(n)$ and $d(n)$ are perfectly correlated such as $x'(n) = h(n) * d(n)$, $C_{dx'}(\omega)$ is 1. Therefore the power of the error $S_{ee}(\omega)$ is 0. This equation indicates that the performance of the canceller system is dependent on the coherence, which is a measure of noise and the relative linearity of the two processes $d(n)$ and $x'(n)$.

To check the influence of error we can replace $x'(n)$ with $x'(n) = \tilde{x}(n) + n(n)$ where $\tilde{x}(n) = h(n) * x(n)$ and $n(n) = n_g(n) + n_q(n) + n_n(n)$. We assume that we have perfect knowledge of $h(n)$. $C_{dx'}(\omega)$ is given by

$$\begin{aligned} C_{dx'}(\omega) &= \frac{|S_{dx'}(\omega)|^2}{S_{dd}(\omega)S_{x'x'}(\omega)} && (14) \\ &= \frac{|S_{d\tilde{x}}(\omega) + S_{dn}(\omega)|^2}{S_{dd}(\omega)\{S_{nn}(\omega) + S_{\tilde{x}\tilde{x}}(\omega)\}} && (15) \\ &= \frac{S_{dd}(\omega)S_{\tilde{x}\tilde{x}}(\omega)}{S_{dd}(\omega)S_{\tilde{x}\tilde{x}}(\omega) + S_{nn}(\omega)S_{dd}(\omega)} && (16) \\ &= \frac{1}{1 + S_{nn}(\omega)/S_{\tilde{x}\tilde{x}}(\omega)} && (17) \end{aligned}$$

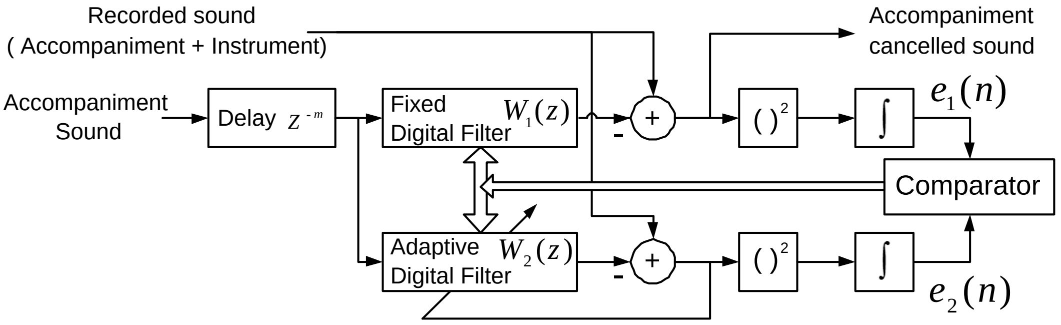
Recorded sound
(Accompaniment + Instrument)
Accompaniment cancelled sound
Accompaniment Sound
Delay z^{-m}
Fixed Digital Filter $W_1(z)$
+
-
( )^2
∫
$e_1(n)$
Comparator
Adaptive Digital Filter $W_2(z)$
+
-
( )^2
∫
$e_2(n)$

Figure 3: Block diagram of Alternative Canceller.

and the power spectrum of the error signal is given by

$$S_{ee}(\omega) = [1 - C_{dx'}(\omega)]S_{\tilde{x}\tilde{x}}(\omega) \quad (18)$$

$$= \frac{S_{nn}(\omega)}{S_{nn}(\omega) + S_{\tilde{x}\tilde{x}}(\omega)} S_{x'x'}(\omega) \quad (19)$$

According this equation, if $S_{\tilde{x}\tilde{x}}(\omega) = 10 \times S_{nn}(\omega)$, the theoretic limitation of the canceller is $\frac{1}{11}$.

After implementing the system as described thus far, we found that sometimes the filter adaptation performs poorly. The adaptation algorithm assumes that the target sound does not exist and that the unwanted sound is white Gaussian. Since these assumptions are not ordinarily true, the adaptation does not always converge to a good estimate of the channel. For example, when the computer-generated sound is very small or silent, it is very unlikely that any changes to the filter will make improvements. Therefore, we developed an extended adaptive method shown by the block diagram in Figure 3.

In this block diagram the cancellation system has two digital filters: an adaptive and a fixed digital filter. The coefficients of the adaptive filter are updated rapidly using incoming samples whereas the coefficients of the fixed digital filter are updated (or not) only at decision points. Between these decision points, we form the sum of error power and decide which filter has better performance. At the end of the decision window (at the decision point), if the adaptive filter performs better, we copy its coefficients to the fixed filter.

4 Implementation

Our ultimate goal is to develop a real-time cancellation system for use in interactive performance. Toward this goal, we implemented the cancellation system as an Aura component and configured a test system using Aura. Aura is a software environment for real-time audio processing; it includes various audio and video signal processing blocks (Dannenberg and Brandt 1996). Using Aura, we implemented the block diagram as shown in Figure 4.

Due to limitations in computation power and also to the difficulty in estimating the high frequency behavior of the channel, we run the cancellation system at 1/4 of the 44,100 Hz sample rate used elsewhere in the system. We put two downsampling blocks and one upsampling block at the ports of the cancellation system. The number of taps (weights) in the canceller is 500, modeling 45.5 ms of the channel’s impulse response. The CPU load is 11% using a 2.4GHz Pentium 4 (and Redhat Linux). The computation time for the cancellation algorithm is $O(n^2)$, where n is the number of filter taps. Therefore, if the sampling frequency is doubled and the time duration of the digital filter is the same, the number of filter taps is doubled, and 4 times the computation power is required.

Notice that latency is independent of CPU load and filter delay. In fact, since the computer “knows” the source of unwanted sound before it even reaches the D/A converters, the cancellation system can estimate the unwanted sound long before it reaches the microphone. The unwanted sound estimate can then be subtracted from incoming samples as soon as they arrive. The only additional delay is due to the downsampling and upsampling filters (see Figure 4). Of course, the computer audio system adds some buffering and therefore latency, but this *system* latency is not increased by the cancellation processing.

For testing, we recorded a stereo waveform in which one channel plays the role of the “computer generated” sound and the other is the “live performer” sound. We play this over two loudspeakers, and place a microphone near the “live performer” loudspeaker to simulate a live performance. This makes the performance repeatable, allowing more controlled experiments. The unwanted sound and the live performance sound are captured and sent to the canceller, which computes the output $y''(n)$. The recorded sound and $y''(n)$ are stored in a sound file in real-time.

To provide one objective measurement of the system per-

Jonghyun Lee, Roger B. Dannenberg, and Joohwan Chun. 2004. “Cancellation of Unwanted Audio to Support Interactive Computer Music.” In *The ICMC 2004 Proceedings*. San Francisco: The International Computer Music Association, pp. 692-698.

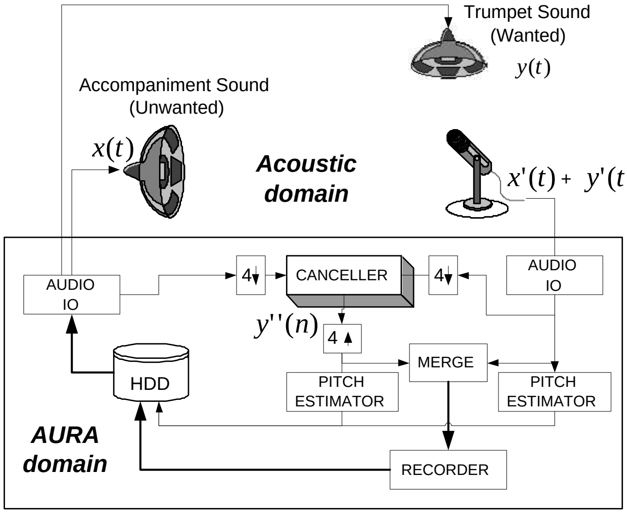

Figure 4: Canceller installed in Aura system.

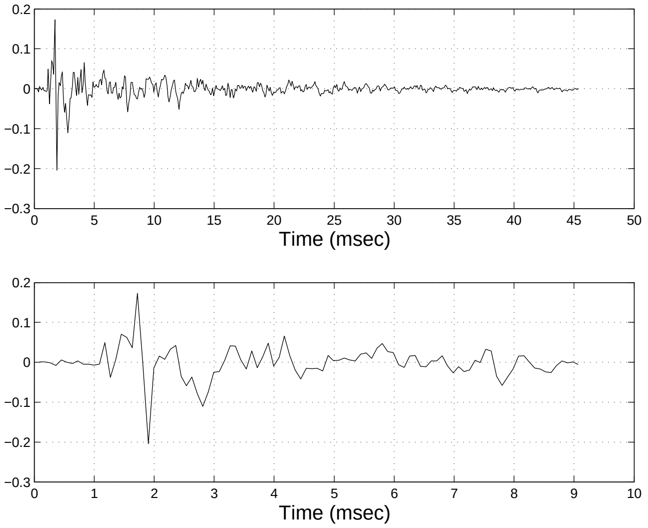

Figure 5: An example of weights.

formance, we connected a pitch estimation[1] module to the output of the cancellation system. Our hypothesis is that by rejecting unwanted sounds from the recording process, we can improve the performance of feature detection such as pitch estimation. The particular pitch estimator uses a very simple time-domain algorithm that relies upon the general shape of trumpet waveforms, which have one pronounced peak per period.

The pitch estimation algorithm generates reports only when consistent consecutive periods are detected. Since the algorithm rarely makes mistakes, a good measure of quality is to count the total number of reported pitch estimates. We perform pitch estimation directly on the recorded sound and also on the output of the cancellation system.

5 Experimental results

The first interesting result is the acoustic channel characteristic, but we cannot find this directly. We only know weights of the digital filter, and we estimate the acoustic channel from the weights. The filter coefficients are highly dependent upon the property of the source, the frequency response of the loudspeaker, the microphone, and other nonlinear effects. Even in the same acoustic channel, the weights depend upon sound sources, and the weights approximate the convolution of the microphone, speaker and channel. An example of weights, $w(n)$, is shown in Figure 5. The top part plots all weights, and the bottom shows only the first 9 ms of the weights. Although it is hard to determine, the duration of the channel appears to be about 35 ms. Assuming that this is an accurate estimate of the impulse response of the channel, it is clear that a simple delay with attenuation would not be a good model of the channel.

We show two types of evaluation results: cancellation performance and pitch estimation performance. Due to the “double-talk” problem mentioned earlier, we do not consider the conventional adaptive filter approach to be suitable for our musical examples, so all of our results are from the complete system (Figure 3) combining an adaptive and a fixed filter. To evaluate the cancellation performance, we set $y(n) = y'(n) = 0$ because we do not have any method to get the waveform $y'(n)$ exactly. Ideally, $y''(n)$ should also be zero, but since cancellation is imperfect, $y''(n)$ will be non-zero. We define the cancellation ratio as

$$CR = \frac{E\{x'^2(n)\}}{E\{y''^2(n)\}} \tag{20}$$

Intuitively, CR is the amount by which the unwanted signal is suppressed (higher is better). The result is shown in the Table 1. Music A is a smooth pop music and music B is hip-hop music with strong percussion sounds. In these tests, CR varies from about 9 to 18dB. Two versions of Music A were tried, one at a sample rate of 11.25kHz and one at 44.1kHz. The CR is better for the 11.25kHz version because the canceller operates at 11.25kHz and therefore high frequencies in the 44.1kHz version are not cancelled. We show an example of $x'(n)$ and the canceller output $y''(n)$ in Figure 6.

Next, we show test results using target sounds. In Figure 7, we show the recorded waveform, cancelled waveform,

[1] Even though technically incorrect, we use “pitch” rather than “fundamental frequency” here because the term is shorter and we feel the meaning is clear in this context.

and target sound $y'(n)$. Here, the “unwanted sound” is a computer-generated music accompaniment containing synthesized piano, bass, and drums playing Gershwin’s jazz standard “Summertime.” The “wanted” sound is a recording of an acoustic trumpet. As explained earlier, these signals exist as left and right channels of a stereo recording. Headphones were used in the original recording process to obtain nearly perfect isolation of the trumpet sound. The channels are played simultaneously (see Figure 4) to test the canceller, simulating a live performance of computer and trumpet. To obtain the “true” value of $y'(n)$ (the trumpet), we can simply run the test again while muting the unwanted channel. Notice that you can still see (and hear) accompaniment sounds in the cancelled sound, but they are much attenuated. For the first 7 seconds of Figure 7 (middle), the system was estimating the delay, and after that time the cancellation algorithm is running.

Listening to the cancelled version of the sound reveals substantial artifacts as the filter coefficients change. As it stands now, this system would not be suitable for reducing cross-talk in a recording for human listening; however, machine listening (or feature extraction) is another interesting possibility that also allows for a more objective evaluation. One useful feature is pitch, and we use performance on pitch estimation to measure the effect of cancellation.

The pitch estimation module uses a very simple algorithm that looks for well-defined pitch periods based on equally-spaced threshold crossings. When detected, pitch and time are logged to a file. The pitch detection performance with and without the canceller is shown in Figure 8. The top graph is obtained from the waveform of recorded sound (without the canceller) and represents performance with perfect cancellation. The other two graphs represent pitch estimation with unwanted sounds added, with and without cancellation.

We use the top graph (no unwanted sound) as a reference to classify points in the other graphs as correct (near a reference point) or incorrect (differing in time by 50 ms and/or pitch by 1 Hz from any reference point). Table 2 shows the number of detected pitches in all three conditions. There are almost 80% more correct pitch estimates using the cancellation system, although the number of incorrect estimates rose from almost none to about 2.5 percent. The canceller apparently removes enough interference to allow the pitch estimation algorithm to detect periodicity at many more places. Note that more sophisticated fundamental frequency estimators might not be so sensitive to interference, so results with other algorithms might vary quite a bit. We believe this test *suggests* that the reduction of unwanted sounds *might* improve audio feature detection. Demonstrating this with a real feature detector in a real performance is left to future work.

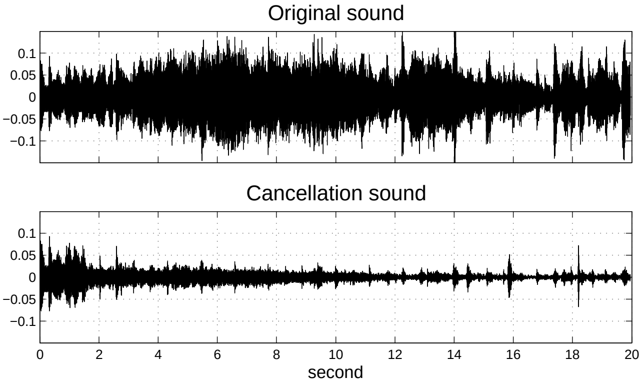

Figure 6: Waveforms without wanted sound.

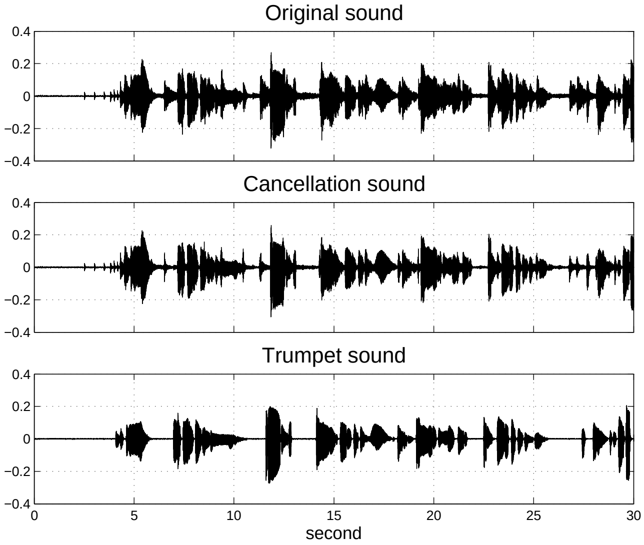

Figure 7: Waveforms with wanted sound.

	$E\{x'^2(n)\}$	$E\{y''^2(n)\}$	CR(dB)
Music A (44kHz)	2.13×10^{-3}	8.46×10^{-5}	14.01
Music A (11kHz)	2.02×10^{-3}	2.81×10^{-5}	18.572
Music B (11kHz)	3.71×10^{-4}	4.27×10^{-5}	9.389

Table 1: Canceling performance.

6 Conclusions

In this paper, a real-time unwanted audio cancellation system is described. The system can be used to enhance the recorded sound’s quality and to improve pitch estimation of a soloist in the presence of computer-generated sound.

The proposed system estimates unwanted audio sounds

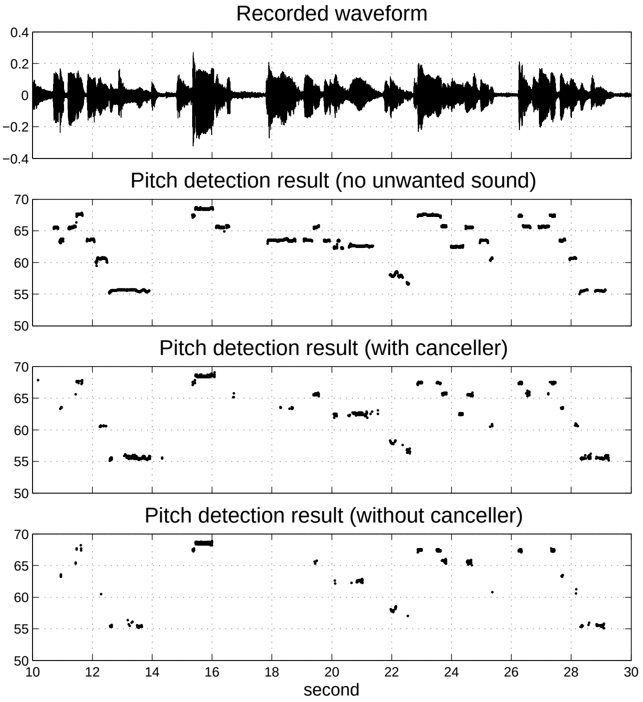

Figure 8: Pitch detection result.

and subtracts them from the recorded sound. Although we know the source waveform of the unwanted audio, the unwanted audio component in the recorded sound is delayed and distorted according to the acoustic channel. We combine a pure delay with a digital filter to compensate for the effect of the acoustic channel and describe the methods of estimating delay and adapting filter weights. To increase the robustness and performance, we developed an alternative adaptation method that avoids making changes that would decrease performance. We implemented the unwanted sound cancellation system in Aura, a real-time platform for interactive computer music.

To demonstrate the cancellation system performance, we show the recorded waveforms and the cancellation ratio (CR). CR depends on the error, especially quantization error. To further evaluate the system, we assembled a pitch estimation application and showed that pitch estimates can be improved when cancellation is applied. (We will play sound examples at the conference.)

	number of wrong points	number of correct points
Instrument only		4829
Instrument with canceller	39	1536
Instrument without canceller	1	859

Table 2: Pitch detecting performance.

In conclusion, unwanted sound cancellation can be applied in real-time to improve the performance of an interactive computer music system. We learned that classical echo cancellation techniques alone are not suitable for this application because of the sparse nature of musical spectra combined with the possibility that the computer-generated sound can contain silence. Our final system combines three components: a channel delay estimator, an adaptive filter, and a controller that ensures that adaptation actually improves performance. All of this runs in real time in software on a single-CPU personal computer. The time-varying nature of the adaptive filter produces artifacts that might be considered more objectionable than the original unwanted sound, so the system should not be used to "clean up" recordings for human listeners. However, the reduction of unwanted noise may be very helpful for various sound analysis tasks. We demonstrated how unwanted sound cancellation can improve the performance of a simple pitch estimation system.

7 Acknowledgements

This work was mainly performed at Carnegie Mellon University by the first author supported by the Brain Korea 21 Project, School of Information Technology, KAIST in 2004 and the Ministry of Science and Technology managed by MICROS and KOSEF (R01-2003-000-10829-0). Additional support (for the second author) came from the National Science Foundation, grant IIS-0085945, and from the Computer Science Department at Carnegie Mellon. This work was inspired by discussions at the Connecticut College Symposium on Art and Technology in 2003 and with Allen Heidorn.

References

Carter, G. C. (1987). Coherence and time delay estimation. *Proceedings of the IEEE 75*, 236–255.

Dannenberg and Brandt (1996, 8). A flexible real-time software synthesis system. In *Proceedings of the International Computer Music Conference*, pp. 270–273. International Computer Music Association.

Haykin, S. (1969). *Adaptive Filter Theory* (3 ed.). Prentice-Hall.

Ianniello, J. P. (1982, 12). Time-delay estimation via cross-correlation in the presence of large estimation errors. *IEEE Transaction on Acoustics, Speech and Signal processing 30*, 998–1003.

Jonghyun Lee, Roger B. Dannenberg, and Joohwan Chun. 2004. “Cancellation of Unwanted Audio to Support Interactive Computer Music.” In *The ICMC 2004 Proceedings*. San Francisco: The International Computer Music Association, pp. 692-698.

J. Benesty, D. R. M. and J. H. Cho (2000). A new class of doubletalk detectors based on cross-correlation. *IEEE Transactions on Speech and Audio Processing 8*, 168–172.

Knapp, C. H. and G. C. Carter (1976). The generalized correlation method for estimation of time delay. *IEEE Transaction on Acoustics, Speech and Signal Processing 24*(3), 320–327.

Kuo, S. M. and D. R. Morgan (1999). Active noise control: a tutorial review. *Proceedings of the IEEE 87*(6), 943–975.

Kuo, S. M. and Z. Pan (1993, 12). Distributed acoustic echo cancellation system with double talk detector. *Journal of the Acoustical Society of America 94*(9), 3057–3060.