

SHORT-TIME TIME-REVERSAL ON AUDIO SIGNALS

Hyung-Suk Kim

CCRMA,
Stanford University
Stanford, CA, USA

hskim08@ccrma.stanford.edu

Julius O. Smith

CCRMA,
Stanford University
Stanford, CA, USA

jos@ccrma.stanford.edu

ABSTRACT

We present an analysis of short-time time-reversal on audio signals. Based on our analysis, we define parameters that can be used to control the digital effect and explain the effect each parameter has on the output. We further study the case of 50% overlap-add, then use this for a real-time implementation. Depending on the window length, the effect can modify the output sound variously, from adding overtones to adding reverse echoes. We suggest example use cases and digital effects setups for usage in sound design and recording.

1. INTRODUCTION

Overlap-add (OLA) methods are widely used in digital audio effects. Examples include time stretching, pitch shifting, phase vocoder, and more complex effects based on the short-time Fourier transform (STFT). [1, 2, 3, 4, 5, 6, 7, 8]. In this paper, we explore a special case of OLA effects termed short-time time-reversal (STTR) — reversal of overlapping short time intervals.

Time reversal is widely used in many fields including acoustics, ultrasound, underwater communications, and biomedical engineering as a method for focusing propagated signals ([9, 10]). Contrarily, it does not seem to be a noticeable topic in the digital audio effects literature. The application of time reversal in audio effects is generally not covered because the system becomes non-causal. For short time intervals, however, it is possible to add a short delay to the output, a buffering period similar to that of delay line effects, to alleviate non-causality. It is worth noting that though STTR is linear, it is not time invariant.

Time reversal audio effects are available on the market. Grain Reverser, a Max for Live plugin, and Reverse Grain from Native Instruments are examples. These audio effects are designed to be temporal not spectral. As we will examine in later sections, time reversal of shorter time intervals, 30 ms or less, with overlap-add creates complex spectral and temporal effects and opens new possibilities for sound design. However, due to the nature of the effect it may be hard to control and may create unexpected and unpleasant results. We shed light on this through Fourier analysis.

The remainder of this paper is structured as follows. We mathematically define STTR and look at the Fourier analysis of STTR (§2), cover the parameters of STTR and examine the effects of each parameter (§3), explore a special case with 50% OLA (§4), look at a real-time implementation of the 50% OLA case (§5), and discuss observations using the implementation (§6).

2. FOURIER ANALYSIS

In this section, we define STTR and perform a Fourier transform to study its effects in the frequency domain.

2.1. Short-Time Time-Reversal

Let $x(t)$ be the input signal and $w(t)$ be the window function of length L with constant overlap-add for step size R : (Equation 2.1)

$$\sum_{m=-\infty}^{\infty} w(t - mR) = 1 \quad (2.1)$$

The STTR signal $y(t)$ is formed by the following steps.

Step 1. Window the input signal $x(t)$ with $w(t - mR)$.

Step 2. Reverse the signal under the window:

- (a) Move the windowed signal to the origin.
- (b) Reverse the windowed signal.
- (c) Move it back to the original position.

Step 3. Sum the reversed signals.

Following the steps we get

$$y(t) = \sum_{m=-\infty}^{\infty} x(-t + 2mR)w(t - mR). \quad (2.2)$$

Note that without the time reversal, the time shifts for $x(t)$ from Step 2 would cancel out.

2.2. General Derivation

The Fourier transform of $y(t)$ becomes

$$Y(f) = \sum_{m=-\infty}^{\infty} e^{-2\pi i f m 2R} X(-f) * e^{-2\pi i f m R} W(f). \quad (2.3)$$

We can expand the convolution in equation (2.3) and use the Fourier transform of an impulse train to simplify $Y(f)$ to

$$Y(f) = \frac{1}{R} \sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{R}\right) W\left(2f - \frac{k}{R}\right). \quad (2.4)$$

For a detailed derivation of equation (2.4), see the Appendix.

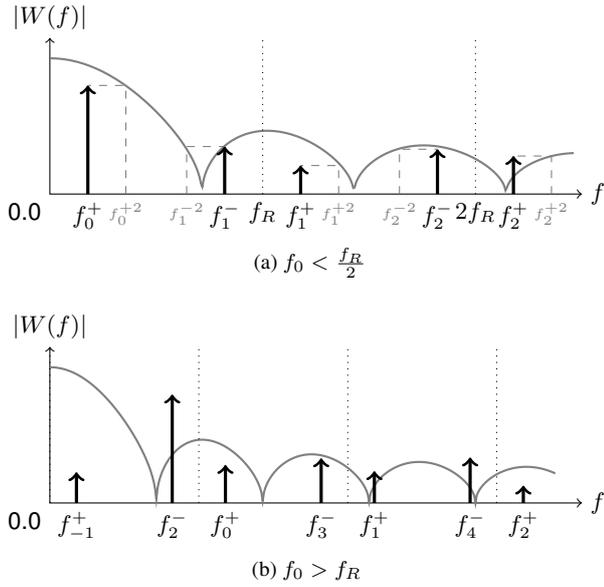


Figure 1: A visualization of equation (2.5), the STTR output for a single sinusoid, for $f_0 < \frac{f_R}{2}$ (1a) and $f_0 > f_R$ (1b). The dotted vertical lines mark multiples of f_R . In 1a, the dashed lines show the magnitude $|W(f_k^{\pm})|$ for frequency f_k^{\pm} . Compared to the window function spectrum $|W(f)|$, the impulse functions need to be scaled by $\frac{1}{2R}$.

2.3. Single Sinusoid Input

From a quick glance equation (2.4) may not intuitively make sense. We can gain insight into the effect of STTR in the frequency domain by looking at the simple case of a single sinusoid.

For a single sinusoid $x(t) = \cos(2\pi f_0 t)$, equation (2.4) becomes,

$$Y(f) = \frac{1}{2R} \sum_{k=-\infty}^{\infty} \left\{ W(f_k^{+2})\delta(f - f_k^+) + W(f_k^{-2})\delta(f - f_k^-) \right\} \quad (2.5)$$

where $f_R = \frac{1}{R}$, $f_k^{\pm} = k f_R \pm f_0$ and $f_k^{\pm 2} = k f_R \pm 2f_0$.

At each integer multiple of frequency f_R , we get two peaks at offsets $\pm f_0$, a weighted copy or “reflection” of the original frequency spectrum (f_k^{\pm}). The weights are given by not the corresponding sample of the window spectrum but that at offsets $\pm 2f_0$, twice the frequency offsets ($f_k^{\pm 2}$). Figure 1 visualizes equation (2.5) for two cases, $f_0 < f_R/2$ and $f_0 > f_R$. When $f_0 < f_R/2$ finding the correct weights for each peak at f_k^{\pm} is trivial. It quickly gets complicated when $f_0 > f_R$ (Figure 1b). Adjacent peaks are not from the same reflection, i.e. f_1^- is not the closest peak to f_1^+ . The weights for each peaks are from even further away points. Furthermore, the original frequency f_0 is not necessarily the peak with the greatest amplitude.

Figure 2b shows the spectrogram of the STTR output for a linear sine sweep for a short window length. The pattern on this figure can be explained by equation (2.5). We cover the parameters of STTR and the observed effects of each parameter in §3.

2.4. Gaussian White Noise

We look at the discrete STTR to analyze the output for Gaussian white noise. Let $x[n]$ be an uncorrelated Gaussian white noise process and $y[n]$ the output after STTR. Since all samples of $y[n]$ are linear combinations of $x[n]$, we know that they are also Gaussian random variables.

We now look at the covariance matrix to verify if all samples in $y[n]$ are uncorrelated. We first look at the case of 50% OLA ($R = L/2$), then extend this to the generalized case. For a given section along the alignment of half the window length, like one slot in Figure 3, the output will be the weighted linear sum of the surrounding time slots. Let X be a $3R \times 1$ random vector with the values of $x[n]$ for $n = [mR, (m+3)R)$, the span of 2 overlapping windows, and Y be a $R \times 1$ random vector with the values of $y[n]$ for $n = [(m+1)R, (m+2)R)$, where the two windows overlap. We can formulate Y as follows,

$$Y = \begin{pmatrix} A & 0 & B \end{pmatrix} X$$

where

$$A_{ij} = \begin{cases} w[j] & i = R - 1 - j; \\ 0 & \text{otherwise,} \end{cases}$$

and

$$B_{ij} = \begin{cases} w[j + R] & i = R - 1 - j; \\ 0 & \text{otherwise.} \end{cases}$$

That is, A and B are cross diagonal matrices with the split window components of $w[n]$ for each overlapping component from the corresponding parts of $x[n]$. Since the covariance matrix of X , V_X is the identity matrix I , the covariance matrix of Y is

$$V_Y = \begin{pmatrix} A & 0 & B \end{pmatrix} I \begin{pmatrix} A \\ 0 \\ B \end{pmatrix} = AA^T + BB^T \quad (2.6)$$

Since A and B are cross diagonal matrices, V_Y is a diagonal matrix and thus all elements of Y are uncorrelated. We can generalize equation (2.6) to any overlap ratio by splitting $w[n]$ onto more cross diagonal matrices. This holds true regardless of the window type. The values of the main diagonal, however, will not be constant ($V_Y \neq I$) but will be dependent on $w[n]$ and the overlap ratio. This means the window type and overlap ratio will be imprinted on the variance for each sample within a given slot. See [11] for an analysis of the effects of OLA on noise.

3. PARAMETERS

Equation (2.5) gives us insight into the parameters that can be used to change the audible effects of STTR. First we can change the window type as well as the length of the window L . Also, we can change R , the step size.

3.1. Window Type

The window type defines the shape of the function $W(f)$. This affects the weights of the overtones. Choosing a window with high sidelobe levels, e.g., a rectangular window, will in general increase the power of the overtones. By smoothly changing the window shape it is possible to shape the overtones. Compared to the sidelobe levels, the mainlobe width has a subtle effect of spreading the peak energy, i.e., the frequency with maximum power over a number of sinusoid peaks. It is worth noting that the peak frequency is not necessarily the original sinusoid frequency (Figure 1b).

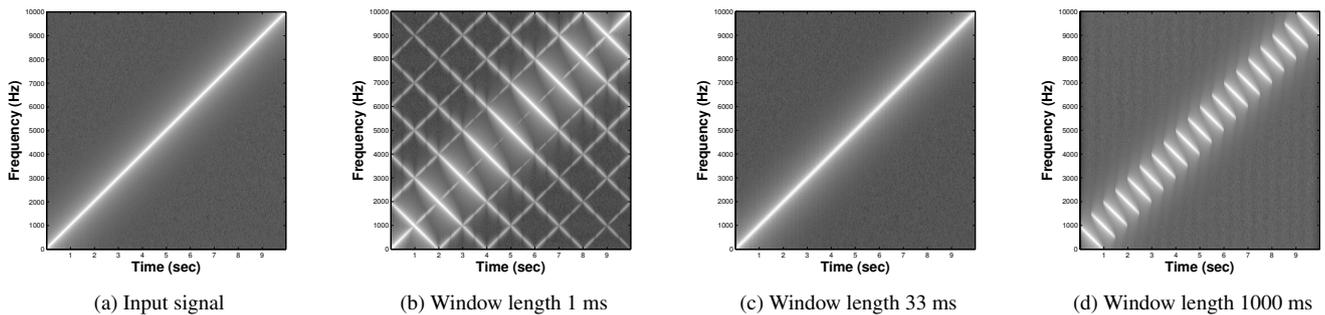


Figure 2: Spectrogram plots showing the effect of STTR window length for 50% overlap-add. The sampling frequency for all signals in this figure is 20 kHz. Figure 2a shows the spectrogram of the input signal, a 10 second linear sine sweep from 0 Hz to 10 kHz. For short window lengths the “reflected” overtones of the signal are visible (2b). As the window length and hop size increase, the reflections are pulled in closer to the main diagonal, decreasing the visibility of STTR on the spectrogram (2c). Further increasing the window length, the time reversal structure becomes visible (2d).

3.2. Step Size

For short window lengths, the step size R changes the reflection frequencies kf_R . Decreasing R will increase spacing between overtones $f_R = 1/R$. The step size will also change the overtone weights as can be seen in Figure 1. We can regulate the overtone weights in regard to the window length by defining the overlap ratio $\alpha = R/L$ and using this as a parameter instead of R .

3.3. Window Length

The window length L determines the width of the window spectrum $W(f)$. As L increases the width of $W(f)$ decreases, eventually resembling an impulse function. At the same time the time reversal effect becomes more audible due to the longer durations that become reversed.

Figure 2 shows the spectrograms of STTR on a linear sine sweep from 0 Hz to 10 kHz with different window lengths. For shorter window lengths (2b), we see the overtones explained in §2.3. For window lengths around 30 ms (2c), the width of $W(f)$ decreases to the point that the reflections disappear on the spectrogram. However, STTR affects the timbre adding roughness or shimmer to the sine sweep. At longer lengths, window lengths beyond 100 ms (2d), we see the overlapping reverse sweeps.

3.4. Relation between Parameters

Though we cover the effects of each parameter separately, it must be noted that they are not independent. The spectrum $W(f)$ depends on both the window type and the window length. The weights of each overtone depend on both $W(f)$ and R .

Furthermore, the step size R must be chosen so that equation (2.1) holds. R cannot be an arbitrary value and is dependent on the type of window as well as its length. When the window side-lobe level is negligible above some frequency f_c , all step-sizes $R < f_s/f_c$ will yield substantially constant overlap-add, where f_s denotes the sampling rate [1, 7].

We can reduce the complexity by fixing the overlap ratio, α . For a fixed α , the window length becomes the parameter that changes the effect of STTR most, since $R = \alpha L$. In the following sections we cover the case where $\alpha = 0.5$ (50% OLA).

4. SPECIAL CASE STUDY: 50% OLA

Here we examine a case for a fixed 50% overlap ratio ($\alpha = 0.5$). The price of fixing α is to lose the freedom of changing the weights of the overtones. However, it simplifies the process of designing a window function.

For a window function to work for 50% OLA, it must satisfy the following constraints.

1. Non-negative $w(t)$ is assumed to be non-negative:

$$w(t) > 0$$

2. Symmetry As with most window functions, we expect $w(t)$ to be even:

$$w(t) = w(-t)$$

3. Constant OLA From equation (2.1) with $C = 1$ and $R = \frac{L}{2}$,

$$w(t) + w\left(\frac{L}{2} + t\right) = 1.$$

From the constraints above, we find that we can choose any shape for the interval $t = [-\frac{L}{2}, -\frac{L}{4}]$, a quarter of the window, with the only constraint being $w(\pm\frac{L}{4}) = \frac{1}{2}$. This opens possibilities for designing various windows to create different overtone weights, including linear mixtures of known constant OLA windows.

5. IMPLEMENTATION

In this section we cover an audio plug-in implementation of 50% OLA STTR. Figure 3 shows the timing relations between the input buffer and the output buffer. This can be implemented efficiently using a single delay line and two output taps. In general, for an arbitrary overlap ratio α , we need $\lceil \frac{1}{\alpha} \rceil$ taps. The length of the delay line is $2L_{max}$, where L_{max} is the longest allowed window length. This is constant regardless of the step size. We can also add another output tap on the delay line to delay the input signal to match that of the STTR signal.

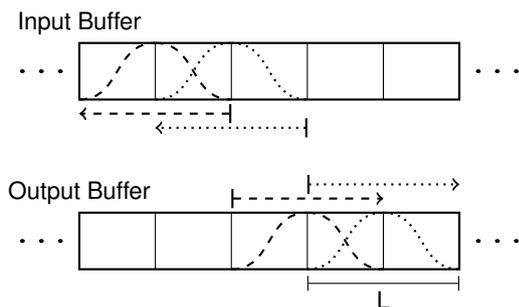


Figure 3: Relation between input signal and output signal for 50% OLA STTR. It shows two overlapping windows and the corresponding read and write directions. We can see that the first sample of a windowed input signal will be the last to be rendered to the output, two window lengths later.

We implemented 50% OLA STTR as an audio plugin with JUCE¹. For practical purposes, we exposed the parameters, window length, window shape and wet/dry mix. The window length parameter is on a log scale ranging from 0.1 ms to 0.5 s. The window shape parameter, ranging from 0 to 1, mixes a rectangular window with a Hann window with 0 being the rectangular window and 1 being the Hann window. The wet/dry mix weighs the output of STTR with the original signal. This is particularly useful for taming the harshness of STTR caused by the overtones. We will look at some example uses in §6.2.

6. OBSERVATIONS

Implementation of an audio plug-in allows the real-time exploration of the digital audio effect. In this section, we test various input signals and present the findings. We cover the perceptual qualities of STTR and suggest example use cases.

6.1. Perceptual Effects of STTR Depending on Window Length

In §2, we covered the effects of window length on a single sinusoid and Gaussian white noise. Here we will make a qualitative assessment on the effects of STTR on more complex audio signals.

For window lengths of less than 1 ms, STTR creates many regularly spaced overtones. This causes the output to sound harsh, metallic and aliased, but with no stretching of the original frequencies. The effects are mostly spectral with almost no effect in the time domain.

For window lengths between 1 ms and 30 ms, we start hearing deflections in the transitions, that is, the pitch, like that of a singing voice, starts moving in a different direction than the original signal. Tonal sounds start sounding detuned.

From 30 ms to 100 ms, the sounds start to flutter. STTR starts having a temporal effect. For sounds like guitar, it adds a shimmering effect, similar to a mixture of chorus and reverb.

Beyond 100 ms, we hear the time reversal. Mixing some of the input signal makes it a reverse echo effect. Due to the delay in the implementation, when mixed with the input signal the delay

becomes noticeable at larger window sizes, which also contributes to the timbre.

6.2. Example Usage

Based on the observations in the previous section we have found example use cases for our implementation of 50% OLA STTR.²

One obvious use is to set a long window length, mix the output with the dry signal and use it to create a reverse echo effect. This works particularly well with arpeggiated instruments such as guitars or pianos.

STTR can be used to change the direction of pitch by setting the window length around 1 ms and 30 ms. Since this extends the spectrum, it is recommended to add a low pass filter to reduce the extreme overtones. This can be used on pitched sounds such as a speech or a car accelerating to make versions with different pitch trajectories.

STTR can also be used to extend the spectrum and add sparkle when set to very short window lengths. For this use, it is recommended to use a low pass filter or band pass filter as an input stage to control the aliasing effects and also a low pass filter on the output stage to reduce extreme overtones.

7. CONCLUSION

We have presented STTR, a novel digital audio effect for manipulating an input signal both spectrally and temporally. Despite its simple implementation, one delay line and a few output taps, it is possible to achieve a variety of effects by changing the window length. STTR opens up new methods for designing and manipulating sounds. We conclude this paper by examining possible extensions of STTR.

We examined the case of 50% OLA STTR and found the degrees of freedom for designing window functions to shape the overtone. It is worth looking further into the effects of the shape of the window function on the timbre and find window design principals for 50% OLA STTR.

Another aspect to further investigate is the effect of time varying window lengths. We hypothesize that for short window lengths, the effect will be similar to a chorus effect (time varying comb filters), yet the spectral peaks will move in alternating directions which may cause a different perceptual effect. We have yet to see what the effect will be at longer window lengths.

Pitch synchronous STTR is also a promising direction to explore. At short window lengths, STTR expands the spectrum of an input signal. Together with a pitch tracker, it may be possible to harmonize a musical signal tonally or atonally. This can also be used to bend the direction of pitch by taking advantage of the fact that we have overtones moving in both directions.

8. REFERENCES

- [1] Jont B. Allen and Larry R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564, Nov. 1977.
- [2] Mark Dolson, "The phase vocoder: A tutorial," vol. 10, no. 4, pp. 14–27, 1986.

¹<http://www.juce.com>

²Sound examples can be found at <http://ccrma.stanford.edu/~hskim08/sttr/>.

- [3] James A. Moorer, “The use of the phase vocoder in computer music applications,” vol. 26, no. 1/2, pp. 42–45, Jan./Feb. 1978.
- [4] Jean Laroche and Mark Dolson, “New phase-vocoder techniques for real-time pitch shifting, chorusing, harmonizing, and other exotic audio modifications,” *Journal of the Audio Engineering Society*, vol. 47, no. 11, pp. 928–936, Nov. 1999.
- [5] Axel Röbel, “A new approach to transient processing in the phase vocoder,” 2003.
- [6] S. Roucos and A. M. Wilgus, “High quality time scale modification for speech,” Mar. 1985, pp. 493–496, Introduces the SOLA (Synchronous OverLap-Add) technique for time scale modification.
- [7] Julius O. Smith, *Spectral Audio Signal Processing*, <http://ccrma.stanford.edu/~jos/sasp/>, Dec. 2011, online book.
- [8] Udo Zölzer, *DAFX*, Digital Audio Effects. John Wiley & Sons, Mar. 2011.
- [9] Sylvain Yon, Mickael Tanter, and Mathias Fink, “Sound focusing in rooms: The time-reversal approach,” *The Journal of the Acoustical Society of America*, vol. 113, no. 3, pp. 1533–1543, 2003.
- [10] Mathias Fink, Gabriel Montaldo, and Mickael Tanter, “Time-reversal acoustics in biomedical engineering,” *Annual review of biomedical engineering*, vol. 5, no. 1, pp. 465–497, 2003.
- [11] Pierre Hanna and Myriam Desainte-Catherine, “Adapting the overlap-add method to the synthesis of noise,” pp. 101–104, 2002.

9. APPENDIX: DERIVATION OF EQUATION (2.4)

$$\begin{aligned}
 Y(f) &= \sum_{m=-\infty}^{\infty} \left(e^{-2\pi i f m 2R} X(-f) * e^{-2\pi i f m R} W(f) \right) \\
 &= \sum_{m=-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-2\pi i (f-\tau) m 2R} X(\tau - f) e^{-2\pi i \tau m R} W(\tau) d\tau \\
 &= \int_{-\infty}^{\infty} X(\tau - f) W(\tau) \left(\sum_{m=-\infty}^{\infty} e^{-2\pi i (2f-\tau) m R} \right) d\tau \\
 &= \int_{-\infty}^{\infty} X(\tau - f) W(\tau) \frac{1}{R} \sum_{k=-\infty}^{\infty} \left(\delta(2f - \tau - \frac{k}{R}) \right) d\tau \\
 &= \frac{1}{R} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} X(\tau - f) W(\tau) \delta(\tau - 2f + \frac{k}{R}) d\tau \\
 &= \frac{1}{R} \sum_{k=-\infty}^{\infty} X(f - \frac{k}{R}) W(2f - \frac{k}{R})
 \end{aligned}$$

On the third line, we use the Fourier transform of an impulse train, the Dirac comb function $\text{III}_T(t)$.

$$\text{III}_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT) = \frac{1}{T} \sum_{k=-\infty}^{\infty} e^{2\pi i kt/T}$$