Extended Convolution Techniques for Cross-Synthesis

Chris Donahue
UC San Diego
cdonahue@ucsd.edu

Tom Erbe
UC San Diego
tre@ucsd.edu

Miller Puckette
UC San Diego
msp@ucsd.edu

ABSTRACT

Cross-synthesis, a family of techniques for blending the timbral characteristics of two sounds, is an alluring musical idea. Discrete convolution is perhaps the most generalized technique for performing cross-synthesis without assumptions about the input spectra. When using convolution for cross-synthesis, one of the two sounds is interpreted as a finite impulse response filter and applied to the other. While the resultant hybrid sound bears some sonic resemblance to the inputs, the process is inflexible and gives the musician no control over the outcome. We introduce novel extensions to the discrete convolution operation to give musicians more control over the process. We also analyze the implications of discrete convolution and our extensions on acoustic features using a curated dataset of heterogeneous sounds.

1. INTRODUCTION

Discrete convolution (referred to hereafter as *convolution* and represented by $*$) is the process by which a discrete signal f is subjected to a finite impulse response (FIR) filter g to produce a new signal $f * g$. If f has a domain of $[0, N)$ and is 0 otherwise and g has a domain of $[0, M)$, then $f * g$ has a domain of $[0, N + M - 1)$. We define convolution as Eq. (1).

$$(f * g)[n] = \sum_{m=0}^{M-1} f[n-m]\, g[m] \tag{1}$$

The convolution theorem states that the Fourier transform of the result of convolution is equal to the point-wise multiplication of the Fourier transforms of the sources. Let $\mathcal{F}$ denote the discrete Fourier transform operator and $\cdot$ represent point-wise multiplication. An equivalent definition for convolution employing this theorem is stated in Eq. (2) and is often referred to as *fast convolution*.

$$\begin{aligned} \mathcal{F}(f * g) &= \mathcal{F}(f) \cdot \mathcal{F}(g) \\ &= \|\mathcal{F}(f * g)\|\ e^{\mathrm{i}\,\angle\mathcal{F}(f*g)} \\ \text{where } \|\mathcal{F}(f * g)\| &= \|\mathcal{F}(f)\| \cdot \|\mathcal{F}(g)\|, \\ \angle\mathcal{F}(f * g) &= \angle\mathcal{F}(f) + \angle\mathcal{F}(g) \end{aligned} \tag{2}$$

When employing convolution for cross-synthesis, one of the two sounds is interpreted as an FIR filter and applied to the other. There are several issues with convolutional cross-synthesis that restrain its musical usefulness.

Treating one of the sounds as an FIR filter essentially interprets it as a *generalized resonator* [1]. However, because convolution is a commutative operation the process is akin to coupling two resonators. The results of convolutional cross-synthesis are often consequently unpredictable and ambiguous. Additionally, there is no way to skew the influence over the hybrid result more towards one source or the other.

Another issue with convolutional cross-synthesis is that the frequency spectra of naturally-produced sounds is likely to decrease in amplitude as frequency increases [2]. The convolution of two such sounds will result in strong attenuation of high frequencies which has the perceived effect of diminishing the "brightness" of the result.

Early attempts to remedy the brightness issue, especially with regard to the cross-synthesis of voice with other sounds (vocoding), involved a preprocessing procedure. A "carrier" sound would be whitened to bring its spectral components up to a uniform level to more effectively impress the spectral envelope of a "modulator" sound onto it [3]. While effective at increasing the intelligibility of the modulator, preprocessing still leaves the musician with limited control over the cross-synthesis procedure and is not particularly generalized.

We introduce an extended form of convolution for the purpose of cross-synthesis represented by $\hat{*}$. This formulation allows a musician to navigate a parameter space where both the perceived brightness of the result as well as the amount of influence of each source can be manipulated. We define extended convolution in its full form as Eq. (3). We will present our justification of these extensions from the ground up in Section 2 and analyze their effect on acoustic features in Section 3.

$$\begin{aligned} \mathcal{F}(f \mathbin{\hat{*}} g) &= \|\mathcal{F}(f \mathbin{\hat{*}} g)\|\ e^{\mathrm{i}\,\angle\mathcal{F}(f \mathbin{\hat{*}} g)} \\ \text{where } \|\mathcal{F}(f \mathbin{\hat{*}} g)\| &= (\|\mathcal{F}(f)\|^{p} \cdot \|\mathcal{F}(g)\|^{1-p})^{2\,q}, \\ \angle\mathcal{F}(f \mathbin{\hat{*}} g) &= 2\,s\,(r\,\angle\mathcal{F}(f) + (1-r)\,\angle\mathcal{F}(g)) \end{aligned} \tag{3}$$

2. EXTENDING CONVOLUTION

In this section we will expand on our extensions to convolution based on the two criteria we have identified: control over the brightness and source influence over the outcome.

2.1 Brightness

Convolution of arbitrary sounds has a tendency to exaggerate low frequencies and understate high frequencies. One way to interpret the cause of this phenomenon is that the

magnitude spectra of the two sounds constructively and destructively interfere with each other when multiplied during convolution. The interference of low-frequency peaks in natural sounds is likely to yield higher resultant amplitudes than the interference of high-frequency peaks.

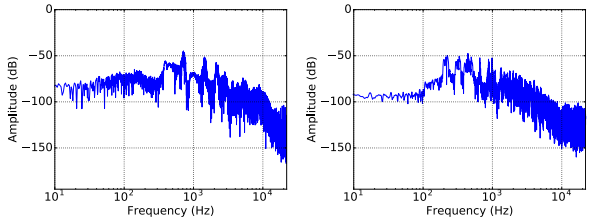
To resolve this issue, we employ the geometric mean when combining the magnitude spectra of two sounds. The geometric mean mitigates both the constructive and destructive effects of interference resulting in a more flattened spectrum. In Eq. (4), we alter the form of the convolved magnitude spectrum from Eq. (2).

$$\|\mathcal{F}(f \hat{*} g)\| = \sqrt{\|\mathcal{F}(f)\| \cdot \|\mathcal{F}(g)\|} \tag{4}$$

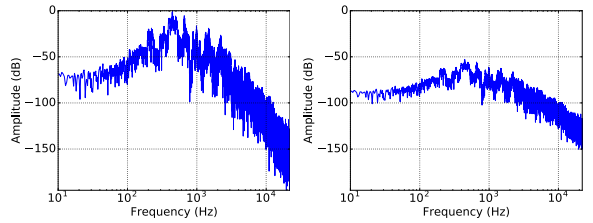
More generally, we introduce a parameter q that controls the flatness of the hybrid magnitude spectrum in Eq. (5).

$$\|\mathcal{F}(f \hat{*} g)\| = (\|\mathcal{F}(f)\| \cdot \|\mathcal{F}(g)\|)^{q} \tag{5}$$

Note that this formulation collapses to *ordinary convolution* as defined in Eq. (2) when $q = 1$ and *geometric mean magnitude convolution* as defined in Eq. (4) when $q = 1/2$. As q decreases towards 0, the magnitude spectrum flattens resulting in noisier sounds. We demonstrate this effect in Figure 1. As q increases past 1, constructive interference between the frequency spectra of f and g is further emphasized, eventually resulting in tone-like sounds.



(a) Source 1 magnitude spectrum (b) Source 2 magnitude spectrum



(c) Ordinary convolution magnitude spectrum ($q = 1$) (d) Geometric mean magnitude convolution spectrum ($q = 1/2$)

Figure 1: Example of cross-synthesis of two sounds (Figure 1a and Figure 1b) using ordinary convolution (Figure 1c) and geometric mean magnitude convolution (Figure 1d).

Eq. (6) is an alternative but equivalent method of calculating $\mathcal{F}(f \hat{*} g)$ with magnitude as defined in Eq. (5) and phase as defined in Eq. (2). It does not use any trigonometric functions and can generally be computed faster in conventional programming environments.

$$\mathcal{F}(f \hat{*} g) = \frac{(a\,c - b\,d) + \mathrm{i}\,(b\,c + a\,d)}{((a^2 + b^2)\,(c^2 + d^2))^{\frac{1-q}{2}}}, \tag{6}$$
$$\text{where } a = \Re(\mathcal{F}(f)),\ b = \Im(\mathcal{F}(f)),$$
$$c = \Re(\mathcal{F}(g)),\ d = \Im(\mathcal{F}(g))$$

2.2 Source Emphasis

We would like the ability to "skew" emphasis of the cross-synthesis result more towards one sound or the other. Our separation of sources into magnitude and phase spectra via fast convolution allows us to modify the amount of influence each source has over the result.

2.2.1 Skewed Magnitude

We extend Eq. (5) to Eq. (7), adding a parameter p which allows the influence of source magnitude spectra $\|\mathcal{F}(f)\|$ and $\|\mathcal{F}(g)\|$ to be skewed in the outcome $\|\mathcal{F}(f \hat{*} g)\|$.

$$\|\mathcal{F}(f \hat{*} g)\| = (\|\mathcal{F}(f)\|^{p} \cdot \|\mathcal{F}(g)\|^{(1-p)})^{2\,q} \tag{7}$$

With this form $p = 1$ fully emphasizes $\|\mathcal{F}(f)\|$, $p = 0$ fully emphasizes $\|\mathcal{F}(g)\|$, and $p = 1/2$ emphasizes neither. As p skews further towards 0 or 1, one of the source's magnitude spectrum is increasingly flattened and the result becomes akin to vocoding. We multiply q by the coefficient 2 to maintain the same scale as in Eq. (5) when $p = 1/2$.

2.2.2 Skewed Phase

We make a similar extension for the phase of the outcome in Eq. (8), adding a parameter r which allows the influence of source phase spectra $\angle\mathcal{F}(f)$ and $\angle\mathcal{F}(g)$ to be skewed in the outcome $\angle\mathcal{F}(f \hat{*} g)$.

$$\angle\mathcal{F}(f \hat{*} g) = 2\,(r\,\angle\mathcal{F}(f) + (1 - r)\,\angle\mathcal{F}(g)) \tag{8}$$

With this form $r = 1$ fully emphasizes $\angle\mathcal{F}(f)$, $r = 0$ fully emphasizes $\angle\mathcal{F}(g)$, and $r = 1/2$ emphasizes neither. We multiply by the coefficient 2 to maintain the analogy of parameter r to parameter p. With the addition of r, the original input sounds can be recovered in the extended convolution parameter space ($p = r = [0, 1], q = 1/2$).

2.3 Phase Scattering

We suggest one final extension to convolution for the purpose of cross-synthesis that does not directly address our two core issues of brightness and source influence. Analogous to parameter q for manipulating the hybrid magnitude spectra, we introduce a parameter s to our definition of hybrid phase spectra in Eq. (9).

$$\angle\mathcal{F}(f \hat{*} g) = 2\,s\,(r\,\angle\mathcal{F}(f) + (1 - r)\,\angle\mathcal{F}(g)) \tag{9}$$

As s decreases towards 0, the source phase is nullified resulting in significant amounts of time domain cancellation yielding impulse-like outcomes. As s increases past 1, the source phase is increasingly scattered around the unit circle eventually converging to a uniform distribution. Randomizing phase in this manner is similar to the additive phase noise of [4] and produces ambient-sounding results with little variation in time.

3. ANALYSIS

In this section we detail our analysis of the effects of convolution and our extended convolution techniques on a curated collection of "random" sounds.

3.1 Data Collection

We used the Freesound API [5] to collect sound material for this research. Our goal is a well-generalized cross-synthesis technique and as such we require a heterogeneous set of sounds for black-box analysis. To achieve this, we made requests to the Freesound API for randomly-generated sound IDs. We used sounds that are lossless, contained one or two channels, had a sample rate of $44.1kHz$, a duration between 0.05 and 5.0 seconds, and were uploaded by a user that was not already represented in the dataset.

We gathered a collection of 1024 sounds satisfying these criteria and henceforth refer to it as the *randomized Freesound dataset* (RFS). We average stereo sounds in RFS to mono and scale all original and hybrid sounds to a peak amplitude of 1 before convolution and analysis.

3.2 Data Preparation

From the 1024 sounds in RFS we generated 512 random pairs of sounds without replacement. We subjected each of these pairs to cross-synthesis via convolution and extended convolution with four different parameter configurations resulting in five sets of 512 hybrid sounds. Using the Essentia software package [6], we perform feature extraction on both RFS and the hybrid sets to analyze what changes convolution yields to acoustic features on average.

We identified the following acoustic features as useful for general analysis: *loudness*, *spectral centroid*, and *spectral flatness*. We computed each of these features for RFS as well as the five hybrid sets listed below. All spectral features were computed using windows of size 1024 with 50% overlap and Hann windowing. Features were averaged across all windows per sound then across all sounds per set.

1. *RFS*: 1024 RFS sounds
2. *OC*: 512 RFS pairs subjected to Ordinary Convolution ($p = 1/2, q = 1, r = 1/2, s = 1$)
3. *GMMC*: 512 RFS pairs subjected to Geometric Mean Magnitude Convolution ($p = 1/2, q = 1/2, r = 1/2, s = 1$)
4. *HPC*: 512 RFS pairs subjected to Half Phase Convolution ($p = 1/2, q = 1, r = 1/2, s = 1/2$)
5. *SMC*: 512 RFS pairs subjected to Skewed Magnitude Convolution ($p = 1, q = 1, r = 1/2, s = 1$)
6. *SPC*: 512 RFS pairs subject to Skewed Phase Convolution ($p = 1/2, q = 1, r = 1, s = 1$)

3.3 Loudness

Raw gain in peak amplitude created by convolving two sources is difficult to predict. Since we are working in the realm of offline cross-synthesis, we ignore the issue and instead focus on perceptual loudness assuming all sounds have been scaled to the same peak amplitude. We hope to use this measure to establish the average effect that convolutional cross-synthesis has on loudness.

Loudness, defined by Steven's power law as energy raised to the power of 0.67 [7], is a psychoacoustic measure representing the perceived intensity of a signal. Loudness of two signals with the same peak amplitude can differ significantly. We calculate loudness for each window of each sound and report loudness for all sets in Table 1.

Set	Mean	Std. Dev.	Min.	Max.
RFS	7.6333	7.8888	0.3095	34.152
OC	7.5306	8.4834	0.0067	41.703
GMMC	5.7503	5.3752	0.3390	28.092
HPC	2.2905	1.9530	0.4358	13.476
SMC	9.6432	9.1378	0.7742	45.805
SPC	6.4163	6.6037	0.6629	40.045

Table 1: Windowed loudness values for all sets.

The mean loudness of all hybrid sets is skewed by the exaggerated tail created by convolution. It is more telling to examine the max loudness. OC produces higher average max loudness than RFS, while GMMC and HPC produce lower max loudness. Both GMMC and HPC have an averaging effect on the amplitude envelope of the result which causes this reduction (as is indicated by their lower standard deviation). SMC and SPC both produce an increase in max loudness compared to RFS that is similar in magnitude to the increase produced by OC.

3.4 Spectral Centroid

The spectral centroid is the barycenter of the magnitude spectrum using normalized amplitude [8]. Listed in Table 2 in Hz, the spectral centroid represents a good approximation of the "brightness" of a sound. The higher the value, the brighter the perceived sound. We use the spectral centroid to quantify our informal observation of high frequency attenuation produced by convolutional cross-synthesis.

Set	Mean	Std. Dev.	Min.	Max.
RFS	3333.3	1467.7	1212.0	7634.0
OC	1206.4	677.18	341.14	4480.5
GMMC	3590.2	745.49	2049.8	6083.7
HPC	1206.6	433.52	803.56	5455.0
SMC	1048.8	351.16	623.38	3842.9
SPC	1156.2	343.68	677.39	3768.9

Table 2: Spectral centroid values for all sets.

The average spectral centroid of the outcome of OC is approximately 64% lower than that of RFS. This confirms our observation that the output of convolution is often perceptually darker than the inputs. GMMC brings the average spectral centroid to a similar level of the original sounds. Both SMC and SPC have a similar effect on the perceived brightness compared to OC, indicating that our source influence parameters (p, r) are relatively independent from our parameter controlling brightness (q).

3.5 Spectral Flatness

Spectral flatness is a measure of the noisiness of a signal and is defined as the ratio of the arithmetic mean to the geometric mean of spectral amplitudes [8]. The measure approaches 1 for noisy signals and 0 for tonal signals. Spectral flatness values for all sets appear in Table 3. We use

this measure to demonstrate our informal observation that ordinary convolution overemphasizes constructive spectral interference yielding results that are less flat than the inputs.

Set	Mean	Std. Dev.	Min.	Max.
RFS	0.2158	0.1132	0.0628	0.5778
OC	0.0207	0.0299	0.0015	0.2761
GMMC	0.2649	0.0670	0.1289	0.5192
HPC	0.0191	0.0550	0.0040	0.6240
SMC	0.0073	0.0320	0.0005	0.4047
SPC	0.0163	0.0310	0.0026	0.3972

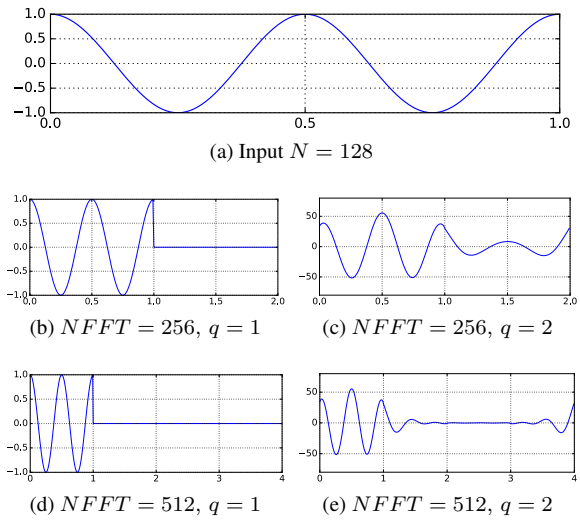
Table 3: Spectral flatness values for all sets.

The difference between the spectral flatness for OC and GMMC is pronounced; the average spectra for the product of GMMC is roughly 13 times flatter than that of OC. SMC further emphasizes constructive interference as it is a self-multiplication and produces even less flat results than OC. HPC and SPC have a less notable effect on spectral flatness as the measure does not consider phase.

4. CIRCULAR ARTIFACTS

Nonlinear manipulation of magnitude and phase components in the frequency domain via extended convolution has particular side effects in the time domain.

In Figure 2 we show a 128-sample sinusoid undergoing convolution with a unit impulse using $q = 1$ (ordinary) and $q = 2$ (squared magnitude). We see that ordinary convolution preserves the precise arrangement of frequency components that allow zero-padding to be reconstructed with the inverse Fourier transform, while squaring the magnitude spectra does not. Instead, the onset shifts circularly and has an unpredictable amplitude envelope preventing us from discarding zero-padded samples.



(a) Input $N = 128$

(b) $NFFT = 256$, $q = 1$ (c) $NFFT = 256$, $q = 2$

(d) $NFFT = 512$, $q = 1$ (e) $NFFT = 512$, $q = 2$

Figure 2: Example of circular phenomena when convolving a sinusoid with a unit impulse using $q = [1, 2]$.

These types of artifacts are always cyclical and often produce musically interesting results. With extreme parameter configurations for extended convolution, the DFT size (which must be greater than or equal to the sum of the input lengths less one) can be reinterpreted as a parameter that affects the length of the result. This can produce a desirable effect of sustained, ambient timbres especially when using high s values. Unfortunately, these artifacts also prevent a real-time implementation of extended convolution using partition methods. This is an area for future investigation but is not critical for the purpose of cross-synthesis.

5. CONCLUSIONS

We have demonstrated an extended form of convolution for offline cross-synthesis that allows for parametrized control over the result. Through our extensions to convolution, a musician interested in cross-synthesis now has control over brightness as well as independent control over source emphasis in both magnitude and phase. We have also shown that our extensions influence acoustic features of hybrid results in a meaningful way. Cross-platform software implementing the techniques described in this paper can be obtained at `http://chrisdonahue.github.io/ject`.

Acknowledgments

This research is supported in part by the University of California San Diego General Campus Research Grant Committee and the University of California San Diego Department of Music. Thanks to the Freesound team for their helpful project, and to the anonymous reviewers for their constructive feedback during the review process.

6. REFERENCES

[1] M. Dolson, *Recent advances in musique concrete at CARL*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 1985.

[2] M.-H. Serra, *Musical signal processing*. Routledge, 1997, ch. Introducing the phase vocoder.

[3] C. Roads, *The computer music tutorial*. MIT press, 1996.

[4] T. Erbe, *PVOC KIT: New Applications of the Phase Vocoder*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2011.

[5] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proceedings of the 21st acm international conference on multimedia*. ACM, 2013, pp. 411–412.

[6] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, "Essentia: An Audio Analysis Library for Music Information Retrieval." in *ISMIR*, 2013, pp. 493–498.

[7] S. S. Stevens, *Psychophysics*. Transaction Publishers, 1975.

[8] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," 2004.