

# Spectral Sound Gap Filling

## Abstract

We present a new method for automatically filling in gaps of textural sounds. Our approach is to transform the signal to the time-frequency space, fill in the gap, and apply the inverse transform to reconstruct the result. The complex spectrogram of the signal is partitioned into separate overlapping frequency bands. Each band is fragmented by segmentation of the time-frequency space and a partition of the spectrogram in time, and filled in with complex fragments by example. We demonstrate our method by filling in gaps of various types of textural sounds.

## 1. Introduction

Automatically filling in gaps of sound and synthesizing textural sounds with similar characteristics to a given input are important in many applications. Since accurate reconstruction of the gap is impossible, the goal of our algorithm is to fill in the signal to produce a perceptually coherent output. In this work we apply techniques used in texture and image synthesis for context-based sound synthesis. We adopt direct image space methods for synthesizing a time varying audio signal by using the complex spectrogram.

The complex spectrogram is an invertible two-dimensional time-frequency representation resulting from the short-time Fourier transform. In each time-frequency coordinate we consider both magnitude and phase, as well as their gradients in time and frequency. Most of the work done in auditory signal processing and scene analysis is based on time-frequency representations that use spectral properties of the signal within time windows. Additional motivation for representing sound in the time-frequency space is that the ear transforms time oscillations into frequency-dependent nerve firings, and that roughly speaking, sound is perceived in the frequency domain. More perceptually motivated, two-dimensional representations that are based on the short-time Fourier transform, such as the mel-frequency cepstral coefficients, are also suitable for our purposes.

Given an input sound signal with a gap as shown on the top left of Figure 1, the result of our gap filling algorithm

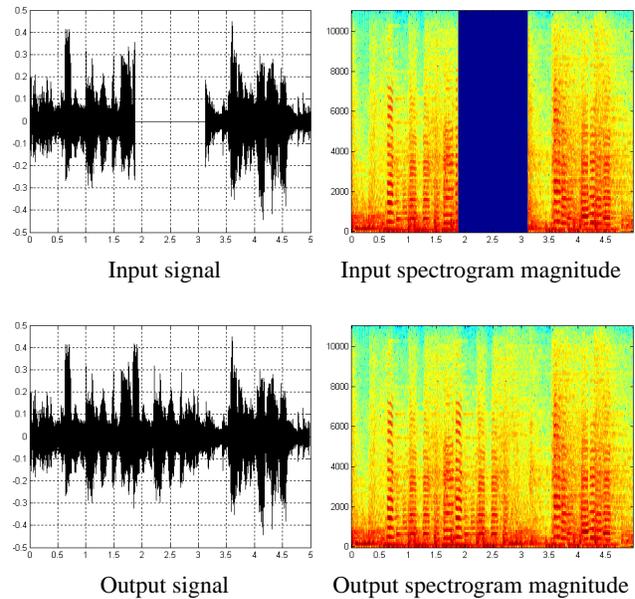


Figure 1. Spectral gap filling of Jazz segment.

is shown on the lower left, and the corresponding spectrogram magnitudes are shown on the right column. The accompanying audio files are available for download from [www.geocities.com/supplemental04audio/icpr](http://www.geocities.com/supplemental04audio/icpr).

### 1.1. Related work

The short-time Fourier transform is a well-established tool used in sound analysis and synthesis [1, 14]. It allows the reconstruction of a signal from its modified short-time Fourier transform [6, 11]. Recently, various algorithms for texture and image synthesis were proposed and applied to the task of filling in missing regions in images. In this work we aim at using similar approaches by transforming the real 1D time signal into a complex 2D time-frequency space. This is performed by matching similar spectral sound fragments and segmentation of the time-frequency space. Ullman et al.[16] emphasize the importance of intermediate-level fragments for the tasks of visual classification and segmentation. Kwatra et al.[13] perform image and video texture synthesis by finding the min-cut of a graph using

a cost function defined on edges between adjacent pixels. Bertalmio et al.[2] combine image inpainting with texture synthesis by decomposing an image into the sum of two components. Inpainting is applied to the component representing the underlying image structure, whereas texture synthesis is separately applied to the component representing image detail, and the two components are then added back together. Fragment-based image completion [7] iteratively approximates the unknown regions and searches for adaptive image fragments under combinations of spatial transformations. Completion is performed from coarse to fine scales, proceeding from regions of high to low confidence. Criminisi et al.[5] fill in an order that gives priority to high gradients. This is achieved in the former by multiplying the traversal map by an adaptive neighborhood size map. Jia and Tang [12] first perform complete segmentation of the input and then continue the segmentation boundaries of the missing regions by tensor voting.

## 2. Time-frequency representation

In this work we use the short-time Fourier transform despite its inherent limitations, namely; a uniform partition of the time frequency plane, with single time resolution for different frequencies (an alternative is a multi-resolution wavelet representation). The advantages of this representation for synthesis are its simplicity - directly synthesizing the complex spectrogram using recent image space techniques, and working with an invertible time-frequency representation that is robust to large modifications [14]. Given a sampled sound signal  $f(t)$ , let  $S_f(x, y)$  denote its discrete short-time Fourier transform, using a Hamming window with an overlap of  $\frac{1}{4}$  window size. The notation  $(x, y)$  represents (time, frequency) emphasizing the image nature of the complex spectrogram. Let  $M(S_f(x, y))$  denote the spectrogram magnitude, and let  $\Phi(S_f(x, y))$  denote the phase. To reconstruct the signal from its complex spectrogram, the inverse short-time Fourier transform is applied to each column, and the overlap-addition method [14, 6] is used to recover the signal. This allows reconstructing spectrograms that have undergone large modifications.

## 3. Sound gap filling

Our approach is to map the signal to the time-frequency space by the short-time Fourier transform, fill in the gap, and apply the inverse mapping to reconstruct the result, as illustrated by:

$$f(t) \mapsto S_f(x, y) \xrightarrow[\text{gap filling}]{} S_{f'}(x, y) \mapsto f'(t) \quad (1)$$

The complex spectrogram of the signal is separated into overlapping frequency bands and partitioned in time. Each

frequency band is filled in by matching fragments and the synthesized result is reconstructed. Gap filling proceeds by matching complex time-frequency fragments to the overlapping regions of existing data from the input and synthesized signal. The criteria for matching fragments is based on magnitude and phase and is performed separately in each frequency band while maintaining coherence between overlapping bands. The complex time-frequency plane is filled in by fragments with adaptive extent in time and frequency. In addition, each fragment forms irregular boundaries in the time-frequency plane within causal neighborhoods, which are determined by local segmentation based on magnitude and phase, and their gradients in both time and frequency. Once the complex time-frequency plane is covered, we apply the inverse transform, and use the overlap-addition method to further blend together the fragments into a coherent output sound stream. Following is a detailed description of each part of our algorithm.

### 3.1. Frequency partition and synthesis order

The complex spectrogram of the signal  $S_f$  is partitioned into separate overlapping frequency bands  $F_k = S_f(:, b_k)$ . Low frequencies of most natural stimuli usually contain more energy than high frequencies and therefore are less affected by noise. Therefore, gap filling of the complex spectrogram proceeds from low to high frequency bands. Perceptual time-frequency representations use a logarithmic frequency scale. Therefore, in our linear frequency scale, the frequency extents are spaced exponentially by multiplying each one from low to high frequencies, such that  $|b_k| = 2|b_{k-1}|$ . Figure 2 shows the spectrogram magnitudes  $M(S_f)$  in consecutive steps of the algorithm for the signal shown in Figure 1. Within each frequency band  $F_k$  we consider fragments  $T_k = S_f(a, b_k)$  that overlap the known regions, and fill in each frequency band from the known to unknown regions of the spectrogram.

### 3.2. Time partition and spectral search

The spectrogram is partitioned in time  $P(S_f)$  by summing the time gradient magnitudes over all frequencies  $\sum_y |\frac{\partial M(S_f(x, y))}{\partial x}|$  for each time  $x$ . At each step of filling in a frequency band  $F_k$ , a target fragment  $T_k$  is defined with a causal region of overlap  $O(T_k)$  with the input and previously synthesized regions. To maintain a coherent sound stream, the time extents  $a = [a_1, a_2]$  of each fragment are determined by the nearest times in the partition of all previously (lower) synthesized frequencies with the greatest response inside the gap. Motivated by our auditory working memory, the maximum overlap in time is 250ms. We search the known complex spectrogram for source matches  $O(T'_k)$  within the same frequency band, across all time in-

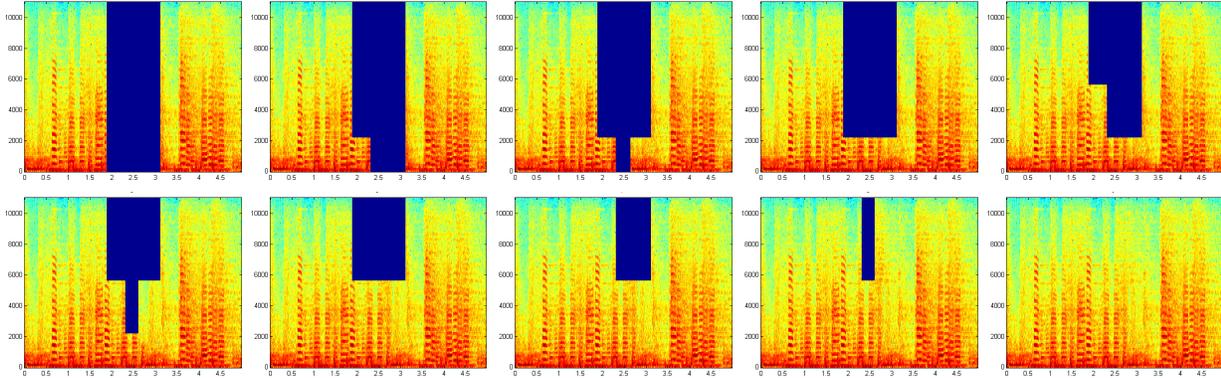


Figure 2. Spectrogram magnitudes in consecutive gap filling steps.

tervals outside the gap. This is a linear one-dimensional search in time which is very efficient. A fundamental task in sound analysis is the comparison of pairs of local spectral representations, and several spectral distance and distortion measures were proposed and analyzed [10]. The amplitude of natural sound signals can rapidly change over several orders of magnitude. Therefore, the similarity of two spectral regions is based on the RMS *logarithmic* spectral distortion [10] used in many speech recognition systems. We have also experimented with the the distortion measure  $\frac{1}{2} - \frac{1}{2} \frac{O(T) \cdot O(T')}{\|O(T)\| \|O(T')\|} \in [0, 1]$  used in indexing [9, 3]. The features are the magnitude and phase  $(M, \Phi)$ , separately normalized taking a weighted average. We find the best overlap match  $O(T'_k)$ , which in turn defines the best fragment match  $T'_k$ . The time extents  $a'$  of  $T'_k$  are similarly updated according to the partition  $P(S_f)$ . The spacing between time extents  $a$  of  $T_k$  inside the gap are set to match corresponding time extents  $a'$  of  $T'_k$  outside the gap, such that  $|a| = |a'|$ , which gives priority to filling the gap with structured sound fragments partitioned in time.

### 3.3. Spectral boundaries

Incrementally, each matching fragment  $T'_k$  fills in a portion of the spectral gap. Its spectral boundaries in the time-frequency space are irregular and based on a local segmentation that determines which disjoint parts to take from  $O(T_k)$  and  $O(T'_k)$ . Locally, the boundaries between fragments define a spectral segmentation with the input and previously synthesized regions. The distortion between spectral features, both magnitude and phase, with priority to high gradient regions of magnitude and phase in time and frequency [8], defines the spectral boundaries which are computed by dynamic programming. The features are magnitude, phase, and their gradients in both time and frequency  $(M, \Phi, \frac{\partial M}{\partial x}, \frac{\partial M}{\partial y}, \frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y})$ . Magnitude and phase information are separately divided by their respective normalized gradient magnitudes in time and frequency, and are

approximated by central and forward differences.

Finally, the output is reconstructed from the filled in complex spectrogram, and the overlap-addition method blends together the fragment boundaries to form a coherent output  $S_{f'}(x, y) \mapsto f'(t)$ .

## 4. Results

We have experimented with our algorithm for gap filling of various types of textural sounds. Computation time is  $O(n \log n)$  in the number of samples  $n$ , and is between 10 and 270 seconds for  $44k$  and  $357k$  samples, on a 1.8Ghz PC processor running Matlab. We use three separate frequency bands  $F_k$ , a Hamming window of length 256 samples, and a gap size of  $\frac{n}{8}$  samples positioned around the  $\frac{n}{3}, \frac{n}{2}, \frac{2n}{3}$  marks. Our algorithm fills in gaps in both abrupt and more continuous textural sounds.

Figure 1 shows the result of filling in a gap of a Jazz segment [15]. Figure 3 demonstrates the result of our gap filling algorithm for various types of sounds. In each row the input signal is shown on the leftmost column, its spectrogram magnitude in the second column, the magnitude of the spectrogram filled in by our algorithm in the third column, and the resulting signal in the rightmost column. The first five rows demonstrate the results of gap filling of natural sounds. The top row shows the result of filling in a gap of a rapidly changing bird song [4], and the second row the result of filling in a gap of the sound of an elk. The third to fifth rows show the results of filling in a gap of a frogs' vocalization [4] for various positions of the gap. The sixth and seventh rows show the results of filling in synthetic sounds of a siren and an engine. Our approach to sound gap filling is example-based and therefore its performance is dependent on the richness of the available fragments. In all the examples presented in this paper, the training set is the known segments of the input, which is rather limited. For example, the last row shows the result of gap filling a musical segment with vo-

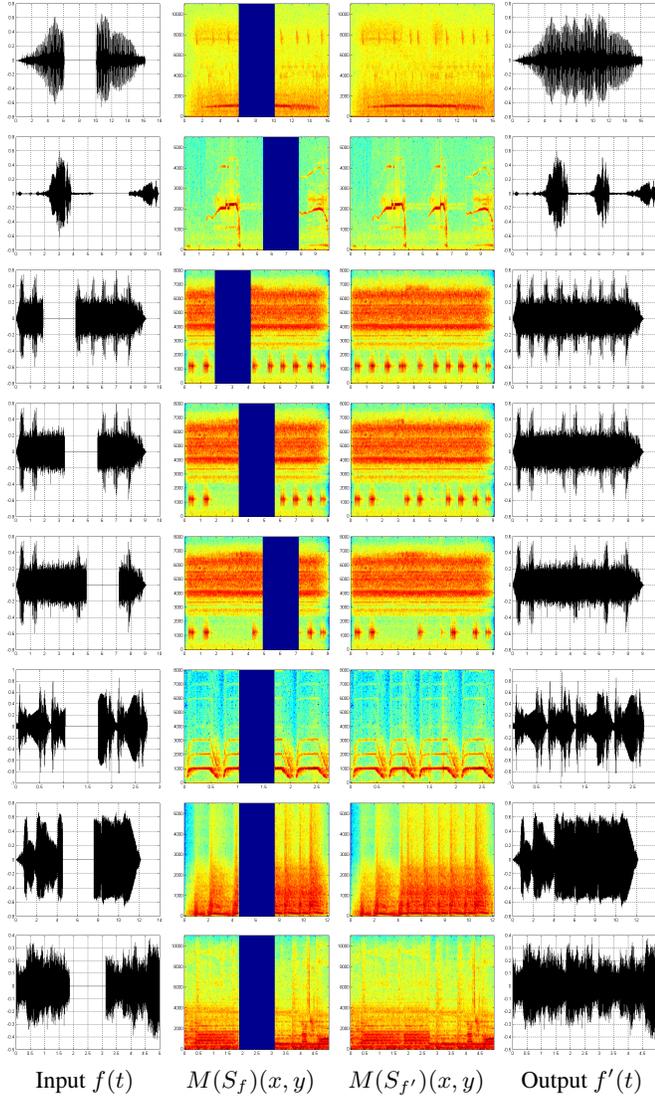


Figure 3. Spectral sound gap filling.

icals [15]. Statistics for each sound in Figure 3 appear in Table 1 (and the accompanying audio files can be downloaded from [www.geocities.com/supplemental04audio/icpr](http://www.geocities.com/supplemental04audio/icpr)).

Sound	Samples (n)	Rate (Hz)	Gap (size,pos)	Time (sec.)
Jazz segment	110250	22050	$\frac{n}{8}, \frac{n}{2}$	57.1
Bird song	356929	22050	$\frac{n}{8}, \frac{n}{2}$	270
Elk	109667	11025	$\frac{n}{8}, \frac{2n}{3}$	35.2
Frogs' vocals	145217	16000	$\frac{n}{8}, (\frac{n}{3}, \frac{n}{2}, \frac{2n}{3})$	55.5, 57.9, 56.9
Siren	43951	16000	$\frac{n}{8}, \frac{n}{2}$	9.9
Engine	134151	11025	$\frac{n}{8}, \frac{n}{2}$	53.7
Music vocals	110250	22050	$\frac{n}{8}, \frac{n}{2}$	65.4

Table 1. Statistics and running times for gap filling of sounds in Figures 1 and 3.

## 5. Future work

To improve the local segmentation we would like to fuse information from multiple candidate target fragments by having each fragment vote on the segmentation. Additional applications include transferring spectral attributes between pairs of signals by constrained synthesis, and extensions of this work to spectral synthesis of image and volumetric data.

## References

- [1] J. B. Allen. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(3):235–238, 1977.
- [2] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.
- [3] W. Chai and B. Vercoe. Structural analysis of musical signals for indexing and thumbnailing. In *Joint Conference on Digital Libraries*, pages 27–34, 2003.
- [4] Cornell Lab of Ornithology. <http://birds.cornell.edu>.
- [5] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 2004, to appear.
- [6] R. E. Crochiere. A weighted overlap-add method of short-time Fourier analysis/synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):99–102, 1980.
- [7] I. Drori, D. Cohen-Or, and Y. Yeshurun. Fragment-based image completion. *ACM Transactions on Graphics (TOG)*, 22(3):303–312, 2003.
- [8] A. Fishbach. Primary segmentation of auditory scenes. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 113–117, 1994.
- [9] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 452–455, 2000.
- [10] R. Gray, A. Buzo, A. Gray, and Y. Matsuyama. Distortion measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):367–376, 1980.
- [11] D. W. Griffin and J. S. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.
- [12] J. Jia and C. K. Tang. Inference of segmented color and texture description by tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, to appear.
- [13] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics (TOG)*, 22(3):277–286, 2003.
- [14] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.
- [15] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proceedings of International Conference on Acoustic, Speech, and Signal Processing*, pages 1331–1334, 1997.
- [16] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):1–6, 2002.