

Deep Audio Waveform Prior

Arnon Turetzky, Tzvi Michelson, Yossi Adi, Shmuel Peleg

School of Computer Science and Engineering
The Hebrew University of Jerusalem

{arnon.turetzky, tzvi.michelson}@mail.huji.ac.il

Abstract

Convolutional neural networks contain strong priors for generating natural looking images [1]. These priors enable image denoising, super resolution, and inpainting in an unsupervised manner. Previous attempts to demonstrate similar ideas in audio, namely deep audio priors, (i) use hand picked architectures such as harmonic convolutions, (ii) only work with spectrogram input, and (iii) have been used mostly for eliminating Gaussian noise [2]. In this work we show that existing State-Of-The-Art (SOTA) architectures for audio source separation contain deep priors even when working with the raw waveform. Deep priors can be discovered by training a neural network to generate a single corrupted signal when given white noise as input. A network with relevant deep priors is likely to generate a cleaner version of the signal before converging on the corrupted signal. We demonstrate this restoration effect with several corruptions: background noise, reverberations, and a gap in the signal (audio inpainting).

Index Terms: deep priors, audio denoising, dereverberation, audio inpainting

1. Introduction

It has been demonstrated that the success of deep convolutional neural networks in computer vision is due to deep priors in the convolutional architecture itself and should not be attributed to the training process alone [1]. Leveraging these priors, it is possible to perform image denoising, super resolution, inpainting, and more in an unsupervised fashion without any pretraining [1, 3]. These discoveries helped bridge the gap between classical methods and machine or deep learning methods, while providing insights into the inner workings of deep architectures.

Zhang et al. [2] showed that classical architectures do not provide good priors for audio. They examined the classical WaveUnet, and showed that even working with spectrograms, regular and dilated convolutions suffer from severe limitations. For example, when trying to fit three noised stationary signals of 1,000, 2,000 and 3,000 Hz the networks fitted the signal and the noise at an equal pace. To overcome these limitations, they proposed the usage of harmonic convolutions. Their goal was to exploit harmonic structures as an inductive bias for auditory signal modeling. Despite the success of their experiments, current SOTA audio architectures do not utilize harmonic structures and tend to rely more on standard building blocks.

Later, Tian et. al. [4, 5] demonstrated that in certain cases deep spectrogram priors can still be found in regular and dilated convolutions. Using varying dilation schedules and interwoven skip connections, Narayanaswamy et al. succeeded in strengthening these priors [6]. Michelashvili et al. [7] showed that it is possible to preform speech denoising in an unsupervised manner without priors by leveraging the inherent lack of structure

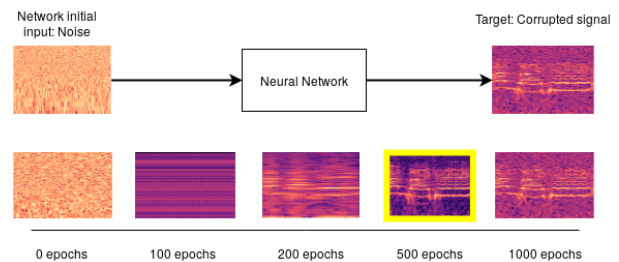


Figure 1: Visualization of the deep prior process for denoising. A Neural Network is trained to generate a target corrupted signal from an initial input which is pure noise. At 500 epochs the generated signal is substantially cleaner than at 1000 epochs, when the corrupted signal is recreated. Network was applied on raw waveform, and spectrograms are used only for visualization.

in noisy signals. In their work, they identified the pixels in the spectrogram which fluctuate most during the training of the network. Using the intensity of the fluctuations they created a mask over the spectrogram which can be used to differentiate between the original signal and the noise.

Recently, Défossez et al. [8] proposed the Demucs model for the task of music source separation. The Demucs model is a neural network composed of a fully convolutional encoder followed by a sequential modeling over the encoder's output, and a reverse decoder with U-net like connections between the encoder and decoder (we describe the model later in more detail). This model reached SOTA performance in music source separation [9, 10] and speech enhancement [8]. In this work we show that using this new architecture it is possible to preform denoising, dereverberation, and inpainting on the raw waveform in an unsupervised manner using priors inherent to the network.

2. Deep Waveform Priors Overview

To discern whether a network contains deep priors relevant to a specific input, we fit a generator network to a single corrupted signal. The network weights are randomly initialized, and fitted to the signal using gradient descent. Thus, the network weights serve as a parametrization for the signal. In this manner, the only information used to perform reconstruction is contained in the single corrupted input signal and the architecture of the generator network. If the architecture of the network contains relevant deep priors, it is possible that during training, the generator will give a cleaner signal before overfitting the corrupted version. Figure 1 presents a diagram showing this process. In the diagram we see that after five hundred epochs the signal is cleaner than the original noisy signal. The graph in Figure 2 shows the behavior of the SI-SNR during the process described above. In the graph, the line representing SI-

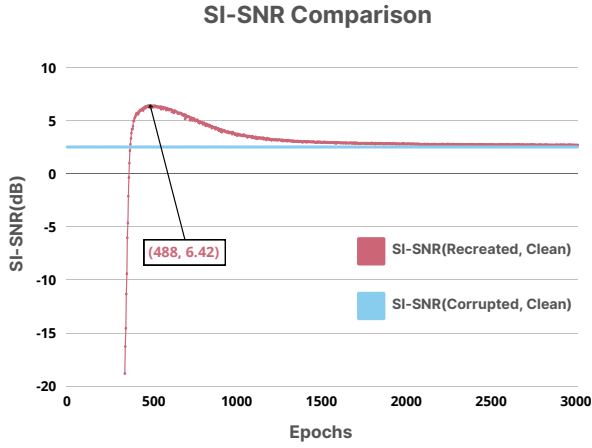


Figure 2: Graph of Scale-Invariant Signal-to-Noise Ratio (SI-SNR) as a function of epochs. We can see that the SI-SNR(Recreated, clean) goes above the SI-SNR(Corrupted, Clean) meaning the recreated signal becomes more similar to the clean signal than the corrupted signal.

SNR(recreated, clean) goes unexpectedly above the line representing SI-SNR(corrupted, clean) before leveling downwards. This essentially means that the recreated signal becomes more similar to the clean signal than the corrupted one.

We use the waveform Demucs architecture [9] as the basis for our analysis. The Demucs architecture consists of a series of down-sampling 1D convolution layers each followed by a Gated Linear Unit (GLU), a number of LSTM layers, and then a series of up-sampling 1D deconvolution layers each followed by a GLU. Following Ulyanov et al. [1] we removed the skip connections from the network. Intuitively, this makes space for the network’s inductive priors to work by propagating the loss only from the very end of the network. When the skip connections remain in place, they correct the weights to overfit the noisy target in every part of the network, and do not leave the network the freedom necessary for the priors to work. In order to allow the network to learn despite the removal of the skip connections it was necessary to reduce the number of convolutional and LSTM layers.

2.1. Implementation details

In different experiments different variants of the architecture were used, to achieve improved results, though a systemic evaluation of all the variants was done only in the ablation studies. The Adam optimizer was used throughout all the experiments with a learning rate of $1e-4$. The networks were trained using L1 loss to calculate the difference from the raw waveforms. The input noise was sampled from a Gaussian distribution with an std of 0.1, and was resampled every epoch in all experiments excluding the audio inpainting, where it was only sampled once. We used a kernel of size 8 with a step of 4 similarly to the default Demucs architecture. In all experiments the networks were trained until convergence (see Figure 2), though almost all converged within 3000 epochs. The audio was sampled at 16Khz in all sections, for all audio types to allow for fair comparison.

2.2. Metrics

We use SI-SNR, PSNR and PESQ as evaluation functions. SI-SNR is defined as in [11, 12] using the following formula:

$$\text{SI-SNR}(s, \hat{s}) = 10 \log_{10} \frac{\|\hat{s}\|^2}{\|e\|^2}, \quad (1)$$

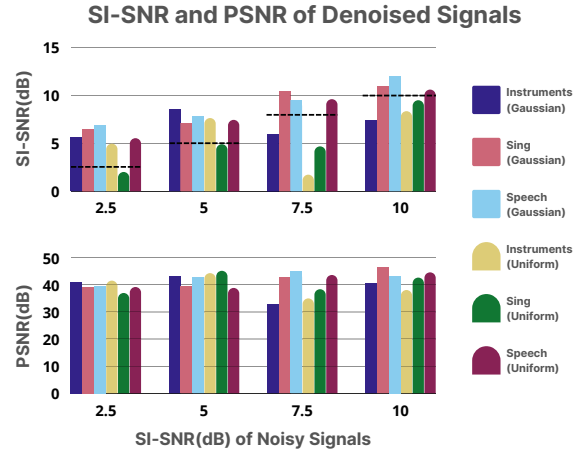


Figure 3: This figure shows the SI-SNR and PSNR of the denoised signals. Each bar represents the average of the metric maximum (similarly to the one marked in Figure 2) over 20 randomly sampled tracks of appropriate class, noise type and noise level. We can see the limits of the deep priors as the SI-SNR increases and it is harder to differentiate between the signal and the noise.

where $\tilde{s} = \frac{\langle s, \hat{s} \rangle s}{\|\hat{s}\|^2}$, $e = \hat{s} - \tilde{s}$. We consider PSNR as follows:

$$\text{PSNR}(s, \hat{s}) = 10 \log_{10} \left(\frac{\text{MAX}_I}{\text{MSE}} \right), \quad (2)$$

where MAX_I is the difference between the maximum and the minimum amplitude values. To calculate PESQ we use the python-pesq package¹.

3. Denoising

The first type of corruption we attempt to correct is denoising. We corrupt the signal using two types of noise:

- **Gaussian noise** — sampled from a Gaussian distribution and added to every sample in the raw waveform
- **Uniform noise** — sampled from a Uniform distribution and added to every sample in the raw waveform

The noise is added to a variety of sounds:

- **Speech** — taken from Valentini et al.’s dataset [13]
- **Singing** — taken from the MUSDB18 dataset [14]
- **Instruments** — taken from MedleyDB 2.0 [15]

Throughout this paper all our clean audio is taken from these 3 datasets. To allow for comparisons between different noise types the noises are added at equal intensities. The addition of the noise was done using code published by Xia et al. [16]. Thus, we create noisy samples with a number of different SI-SNRs and allow comparison between them. The results of our analysis can be found in Figure 3. The graphs report the average maximum metrics achieved by the output signal while training the neural network to generate the corrupted signal. While these results are not competitive with SOTA denoising methods, they do show the existence of a deep prior inherent to the architecture which is the goal of this study. For every level of noising, that is SI-SNR(corrupted, clean), we can see that the network usually performs some denoising before fitting the noisy signal. Figure 4 visualizes one of our denoising results using spectrograms.

¹<https://github.com/ludlows/python-pesq>.

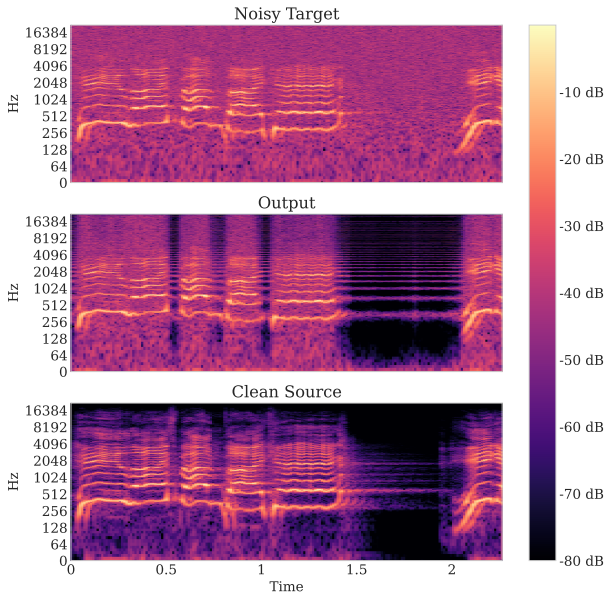


Figure 4: The spectrograms of (a) Noisy target; (b) Network output; (c) Original clean source. As the network recreates the raw waveform, spectrograms are used only for visualization.

4. Dereverberation

Reverberations are acoustical noise appearing in enclosed spaces through multiple reflections of the sound on the walls and objects of a room. When a speaker talks in a room, these multiple echoes add to the direct sound and blur its temporal and spectral characteristics [17]. Reverberant speech can be described as sounding distant with noticeable echos. These detrimental perceptual effects generally increase with increasing distance between the source and the microphone [18]. In this work we attempt to clean the reverberations from the signal treating them as we treated other types of noise in the previous section. This task is inherently more difficult than the other denoising tasks we performed since the corruption itself has the same structure as the signal to be cleaned. As expected, our success here was more limited although a modest amount of dereverberation was obtained. A summary of our experimental results can be found in Table 1. We used the pyroomacoustics python package to add reverberations to our audio [19] as well as Valentini et al.’s dataset [13]. Every setting in the table was averaged over 20 samples, although the results were robust even when averaging over only 10-15 samples.

To better evaluate the level of dereverberation improvement, results reported in Table 1 are the SI-SNR improvement metric (SI-SNRi) over the reverberant signal. We can see that the priors succeed in removing reverberations to a certain extent. PESQ is not reported in the table as it was very unstable and with most samples achieved very high scores which did not represent an improvement in audio quality. Our hypothesis to explain the results in the table is as follows. When the reverberation is very weak ($Rt60=0.1$) the audio sounds very similar to the original, so there is very little to improve, as such the improvements achieved are minor. When the reverberations are of medium strength ($Rt60=0.2$) there are more reverberations to clean, yet they can still be distinguished from the original signal, so the network has a stronger effect. When $Rt60=0.5$ the reverberations are too strong for the prior to differentiate be-

Table 1: Dereverberating results for different input signals with different $Rt60$ s. $Rt60$ is defined as the number of seconds it takes for the reverberations to decrease by 60 dB. The numbers reported in the table are the average SI-SNR and PSNR improvement over the mixture that the network’s output managed to achieve relative to the reverberated signal.

RT60(sec)		SI-SNRi	PSNRi
0.1	Speech	2.22	13.45
	Singing	1.15	14.40
	Instruments	1.15	14.14
0.2	Speech	4.76	16.30
	Singing	5.48	18.45
	Instruments	5.79	18.45
0.5	Speech	-0.11	5.66
	Singing	-0.08	6.65
	Instruments	-0.03	6.45

tween them and the original signal, and the network recreates them both equally well.

5. Audio Inpainting

Inpainting is a classical problem in computer vision and has been used to demonstrate deep priors of convolutional networks. When performing inpainting a mask is placed over part of the image and the network recreates the image behind the mask. In audio, speech and music signals are often subject to localized distortions, where the intervals of distorted samples are surrounded by undistorted samples. Examples may include noises or clicks, CD scratches, old recordings, packet loss in cellphones etc. [20]. Audio inpainting in the waveform domain contains two additional challenges:

- **Dimensionality** – at a standard sample rate of 16KHz masking even 10 ms affects 160 samples. Hence, the network must recreate these 160 consecutive samples.
- **2D vs. 1D** – When recreating an image context can be taken from 360 degrees, as the picture is 2-dimensional. When recreating a waveform, context can only be taken from a single dimension.

In this work we masked a small (1-5 ms, see Table 2), non-silent, segment of a 2 second clip, and trained the network to recreate the input signal. The training loss ignored the network’s output under the mask. Hence, the network is unobstructed in recreating this part of the signal and it’s deep priors can come into view. A visualization of our results can be found in Fig. 5. The figure shows that the network recreates a good approximation of the original signal, despite no loss being calculated on the masked area.

To measure the impact of the network priors we compare the recreation quality of the Demucs network to the recreation quality of WaveUnet which has been shown to contain very weak priors [7, 2]. A summary of the results can be found in Table 2. The results reported are averaged over 20 clips each one of 2 seconds. The masked segment is randomly sampled within the clip. The table shows a number of points. First, the Demucs architecture consistently succeeds in recreating the signal better than the WaveUnet. Second, as the length of the masked signal increases the quality of the recreation goes down. Third,

Table 2: The table describes the metrics reached by the network when inpainting the masked signal using the Demucs architecture vs. using the WaveUnet architecture (which contains substantially weaker priors). Every result is averaged over 20 clips with the masked section randomly sampled within the clip. We can see that the Demucs architecture is consistently superior to WaveUnet in all metrics. However, as the length of the mask grows the quality of the inpainting decreases.

		SI-SNR		PSNR	
		WUnet	Demucs	WUnet	Demucs
1ms	Speech	-2.05	6.57	12.02	32.71
	Music	-1.83	6.81	11.97	34.46
	Singing	-3.73	4.21	11.95	38.40
2ms	Speech	-5.33	3.55	10.53	34.02
	Music	-5.77	2.98	10.79	30.44
	Singing	-8.45	-2.35	10.02	31.64
5ms	Speech	-12.13	-2.60	9.73	30.46
	Music	-9.48	-2.11	10.47	30.03
	Singing	-11.07	-4.60	10.15	31.46

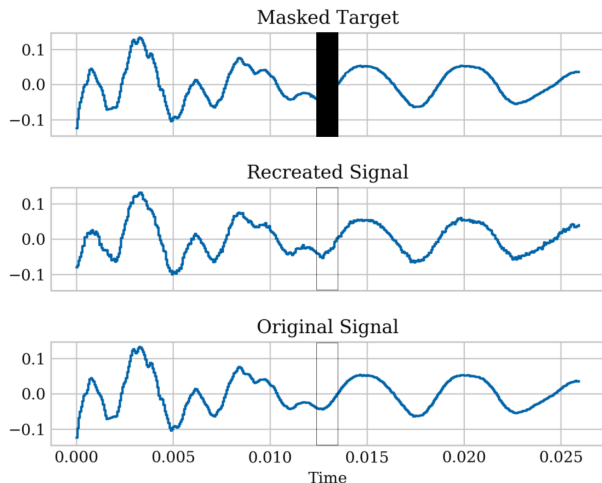


Figure 5: Waveforms of masked signal, recreated signal and original (un-masked) signal. The masked area (the horizontal line in the circle) is recreated and the waveform of the recreated signal is very close to the waveform of the original signal.

the Demucs architecture is less successful at recreating samples of singing. It is interesting to note that the Demucs architecture achieved SOTA results in music separation and speech enhancement, but has not been used for singing audio, similar to our results.

6. Ablation Studies

To better understand which parts of the Demucs architecture cause the architecture to have deep priors we analyzed the network from the bottom up. At each step we added another element to the network and saw how this affects the prior. Our analysis was done using randomly sampled Uniform noise with an SI-SNR of 2.5dB to allow the differences between the variants of the architecture to be seen clearly. With each variant

Table 3: Ablation studies done on different variations of the Demucs architecture. When no activation function is mentioned a standard ReLU is used.

Architecture	SI-SNR	PSNR	PESQ
Conv (2 layers)	3.05	39.12	1.38
Conv (4 layers)	3.07	37.29	1.76
Conv (6 layers)	2.64	38.36	1.33
Conv + skip	2.97	36.07	1.61
Conv + GLU	2.85	36.54	1.42
Conv + LSTM	5.42	40.56	2.31
Conv + Attention	3.26	40.37	1.99
Conv + GLU + LSTM	5.01	39.64	1.85
Conv + GLU + Attention	2.94	37.38	1.96

of the architecture we randomly sampled 20 audio clips, denoised them, and averaged the results. Table 3 reports our results. When the activation is not reported a standard ReLU is used. The original architecture is the Conv + LSTM + GLU option, we added attention layers instead of LSTM layers as part of our ablation studies to see their possible effect. Additionally, we examined the effects of different amounts of convolutional layers as well as the effect of skip connections. Using an LSTM alone did not provide results so the option is not reported in the table. There are a number of points we can learn from the table:

- GLU improves the priors of the network when used with convolutional layers.
- When used with the LSTM layers the effect of the GLU is detrimental.
- Adding LSTM layers to the convolutional layers improves the network’s prior.
- Adding attention layers instead of LSTM layers does not improve the network’s priors.

We understand these results to mean that convolutional layers, aside from containing deep priors themselves, are necessary to prepare the waveform for the sequential modeling (LSTM) and to revert the sequential model back to the waveform. However, the LSTM itself contains significant deep priors. Additionally, although this is not represented in the table, Conv + LSTM works better with only 2 layers of convolution, and not with four. We hypothesize, that as the network deepens the learning becomes harder due to vanishing gradients. Skip connections can not be used as a solution, since they do not leave the network sufficient freedom to learn.

7. Conclusions

In this work we show for the first time (to the best of our knowledge) the power of deep waveform priors in a SOTA audio architecture. We demonstrate the strength of these priors on 3 separate tasks: audio denoising, audio dereverberation, and audio inpainting and achieve results which could not be achieved without deep priors. We believe, the findings presented in this study shed light on the recent success of the Demucs architecture [9] on several source separation tasks operating over the raw-waveform.

Acknowledgements: This research was supported by grants from the Israel Science Foundation, the DFG, and the Crown Family Foundation.

8. References

- [1] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *CVPR*, 2018, pp. 9446–9454.
- [2] Z. Zhang, Y. Wang, C. Gan, J. Wu, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, “Deep audio priors emerge from harmonic convolutional networks,” in *Int. Conf. Learning Representations (ICLR)*, 2019.
- [3] Y. Gandelsman, A. Shocher, and M. Irani, ““double-dip”: Unsupervised image decomposition via coupled deep-image-priors,” in *CVPR*, 2019, pp. 11 026–11 035.
- [4] Y. Tian, C. Xu, and D. Li, “Deep audio prior,” *arXiv:1912.10292*, 2019.
- [5] —, “Deep audio prior: Learning sound source separation from a single audio mixture,” *CVPR Workshops*, 2020.
- [6] V. S. Narayanaswamy, J. J. Thiagarajan, and A. Spanias, “On the design of deep priors for unsupervised audio restoration,” *arXiv:2104.07161*, 2021.
- [7] M. Michelashvili and L. Wolf, “Speech denoising by accumulating per-frequency modeling fluctuations,” *arXiv:1904.07612*, 2019.
- [8] A. Défossez, G. Synnaeve, and Y. Adi, “Real time speech enhancement in the waveform domain,” in *Interspeech*, 2020.
- [9] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music source separation in the waveform domain,” *arXiv:1911.13254*, 2019.
- [10] A. Défossez, “Hybrid spectrogram and waveform source separation,” *arXiv:2111.03600*, 2021.
- [11] E. Nachmani, Y. Adi, and L. Wolf, “Voice separation with an unknown number of multiple speakers,” in *Int. Conf. Machine Learning (ICML)*, 2020, pp. 7164–7175.
- [12] S. E. Chazan, L. Wolf, E. Nachmani, and Y. Adi, “Single channel voice separation for unknown number of speakers under reverberant and noisy settings,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3730–3734.
- [13] C. Valentini-Botinhao, “Noisy speech database for training speech enhancement algorithms and tts models,” University of Edinburgh. School of Informatics. Centre for Speech Technology Research, Tech. Rep., 2017.
- [14] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “The MUSDB18 corpus for music separation,” Dec. 2017. [Online]. Available: <https://doi.org/10.5281/zenodo.1117372>
- [15] R. Bittner, J. Wilkins, H. Yip, and J. Bello, “Medleydb 2.0: New data and a system for sustainable data collection,” in *Int. Conf. on Music Information Retrieval (ISMIR-16)*, 2016.
- [16] Y. Xia, S. Braun, C. K. A. Reddy, H. Dubey, R. Cutler, and I. Tashev, “Weighted speech distortion losses for neural-network-based real-time speech enhancement,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 871–875.
- [17] K. Lebart, J.-M. Boucher, and P. N. Denbigh, “A new method based on spectral subtraction for speech dereverberation,” *Acta Acustica united with Acustica*, vol. 87, no. 3, pp. 359–366, 2001.
- [18] E. Habets, “Single- and multi-microphone speech dereverberation using spectral enhancement,” Ph.D. dissertation, Technische Universiteit Eindhoven, 2007.
- [19] R. Scheibler, E. Bezzam, and I. Dokmanic, “Pyroomacoustics: A python package for audio room simulation and array processing algorithms,” *ICASSP*, Apr 2018.
- [20] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “Audio inpainting,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 922 – 932, Mar. 2012.