AUDIO INTERPOLATION AND MORPHING VIA STRUCTURED-SPARSE LINEAR REGRESSION

Corey Kereliuk
McGill University
CIRMMT†
corey.kereliuk@mail.mcgill.ca

Philippe Depalle
McGill University
CIRMMT†
depalle@music.mcgill.ca

Philippe Pasquier
Simon Fraser University
SIAT*
pasquier@sfu.ca

ABSTRACT

We present a method of audio interpolation suitable for the restoration of missing and/or corrupted audio samples. Our method assumes that the missing/corrupted samples can be easily identified and are subsequently treated as missing data. We then model the audio signal as a linear combination of elementary waveforms (referred to as atoms) and estimate the values of the missing samples by solving a penalized linear regression problem. A first work in this direction was recently presented using the moniker 'audio inpainting' (in deference to similar work in the image processing community). We extend this avenue of research by incorporating additional continuity constraints into the problem, which leads to improved estimates of the missing data. Furthermore, we show how our method leads to a natural framework for morphing/transitioning between two sounds. Finally, we present several examples that illustrate the effectiveness of our interpolation strategy and the quality of morphing that can be attained.

1. INTRODUCTION

It is not uncommon for audio signals to suffer some form of degradation during the various stages of recording, transmission, and playback. For example, a scratched compact disc or dropped network packet can lead to chunks of missing samples. Likewise, impulsive clicks, clipping, and noise are common forms of audio degradation. In this work we focus specifically on localized types of distortion. In other words, we assume that the distorted samples are surrounded by undistorted ones (which occurs in many practical situations). Furthermore, we assume that the distorted samples can be easily identified, either manually, or through some other process (e.g., by detecting regions of silence, clipping, and so on). The task at hand is then one of interpolation, i.e., we aim to estimate the missing samples at known locations using the surrounding data.

There are several works in the literature aimed at audio interpolation [1–5]. In Janssen et al. the sound was modelled as an autoregressive (AR) process and the unknown parameters are estimated from the known data [1]. The missing/corrupted samples can then be interpolated using the AR model.

The work presented in Adler et al. [5] was inspired by related research in the image processing community on 'inpainting' (a process whereby missing pixels are interpolated from the surrounding ones [6]). We may also view the inpainting problem as a regression problem when the signal is modeled as a linear combination of elementary functions (atoms).

In the following sections we describe an extension of the audio inpainting work in [5] using recent results in structured sparse modelling of audio [7]. We compare our approach to those presented in [5] and [1] and show that in both cases we achieve superior reconstruction of the missing samples (in terms of minimizing the estimation error). We also demonstrate how the proposed method leads to a natural framework for morphing/transitioning between two sounds.

The remaining sections are laid out as follows. We first present our model and then outline a sketch of the interpolation problem. We then propose an estimation algorithm based on penalized linear regression with continuity constraints. Finally, we present several examples illustrative of our interpolation and morphing results.

2. THE ADDITIVE MODEL

We adopt the following additive sound model

$$y(t) = \sum_m \sum_n x_{m,n} \phi_{m,n}(t) \tag{1}$$

In other words, the audio signal $y(t)$ is modelled as a linear combination of elementary waveforms $\phi_{m,n}(t)$ referred to as atoms. The double indices (m, n) typically have a time-frequency interpretation (and their extent depends on the signal length and bandwidth). For example, we use Gabor atoms

$$\phi_{m,n}(t) = h(t - am)\exp(j2\pi bnt) \tag{2}$$

which are generated by translating and modulating a smooth and compact window function $h(t)$. In this model the parameters a and b are the time and frequency sampling intervals, m and n are integer indices, $j = \sqrt{-1}$ is the imaginary unit, and t is time. Gabor atoms are a natural choice for modelling audio since they have a compact time-frequency footprint (and thus represent distinct elements

†Centre for Interdisciplinary Research in Music Media and Technology
*School of Interactive Arts + Technology

of the time-frequency plane). The synthesis coefficients $x_{m,n}$ in this model can be calculated using the short-time Fourier transform (STFT). However, when the set of atoms is redundant (e.g., when the Gabor atoms are oversampled in time and/or frequency) there is, strictly speaking, no unique way to determine the synthesis coefficients. In fact, in recent years considerable effort has been invested into examining alternative methods for estimating the synthesis coefficients. In particular, sparse representations are increasingly seen in the literature (see [8] for a review). There are many advantages to sparse models, chiefly data reduction and increased salience of the model parameters (i.e., there is a clearer correspondance between the atoms and the sound signal). Furthermore, many natural signals are inherently sparse which has spurred on much of the growth in this area. For example, in the field of compressive sensing it has been shown that sparse signals can be reconstructed using a small number of measurements [9]. We show in the following sections how sparsity can be used in a similar way to regularize the interpolation procedure.

3. PROBLEM FORMULATION

3.1 Interpolation

In discrete-time the sample values are known at a distinct set of locations and thus the model in Eq. (1) can be rewritten as

$$\mathbf{y} = \mathbf{\Phi x} \tag{3}$$

where $\mathbf{y}$ is a vector containing the audio samples, the columns of the dictionary $\mathbf{\Phi}$ are Gabor atoms, and $\mathbf{x}$ is a vector of synthesis coefficients. When the signal is degraded/distorted we lose information about $\mathbf{y}$. When interpolating audio samples we assume that only some of the samples from $\mathbf{y}$ are reliable and that the others should be treated as missing data (to be re-estimated). We can model this scenario as

$$\mathbf{z} = \mathbf{My} \tag{4}$$

where $\mathbf{z}$ is the observed signal and $\mathbf{M}$ is a diagonal (binary) mask matrix that indicates which samples from $\mathbf{y}$ are reliable and which should be treated as missing data. Replacing $\mathbf{y}$ in Eq. (4) with the model from Eq. (3) leads to

$$\mathbf{z} = (\mathbf{M\Phi})\mathbf{x} = \mathbf{\Psi x} \tag{5}$$

where we use $\mathbf{\Psi}$ to represent the degraded dictionary. If we can accurately estimate $\mathbf{x}$ from $\mathbf{\Psi}$ and $\mathbf{z}$, then we may reconstruct the missing samples via linear regression, i.e., using the linear model in Eq. (3). This summarizes the interpolation setup, however, we have not yet considered how to estimate $\mathbf{x}$. We withhold this discussion until Sec. 4.

3.2 Morphing/Transitioning

We can use the same setup described in the previous section to morph/transition between different sounds. For example, we can generate a new sound by concatenating a source sound, silence, and a target sound together. We may then treat the samples between the source and target as missing data (by generating an appropriate mask matrix). In this case performing the interpolation procedure will create a morph or transition between the two sounds.

We note that this type of morphing is based on waveform interpolation as opposed to feature (or descriptor) interpolation [10]. Descriptor interpolation (e.g., interpolating between partials [11]) is more common in the literature, however, recent examples of waveform interpolation can be found as well. For example in Olivero et al. [12] the authors examined how to find a time-frequency multiplier capable of transforming one sound into another.

Many morphing techniques aim to create several hybrid sounds lying somewhere between the source and target [13]. Our approach, on the other hand, is a simple technique for smoothly transitioning between two sounds and, in this sense, bears more similarity to a cross-fade. However, our morphing results are quite audibly different from cross-fading in many cases (as we demonstrate in the results section). In essence we propose to use (or maybe more accurately abuse) the interpolation procedure in order to produce large chunks of new samples based on the surrounding data.

4. ESTIMATION PROCEDURE

4.1 Penalized linear regression

As we noted in Sec. 2, when the Gabor atoms are oversampled in time and/or frequency (which is typically the case), there is no unique way to determine the synthesis coefficients. Furthermore, even if the Gabor atoms were critically sampled, the degraded dictionary would still be rank deficient due to the multiplication by $\mathbf{M}$ (which discards data). This means that the system of equations in Eq. (5) is underdetermined and there is no unique way to determine $\mathbf{x}$. In this case, we can regularize the problem by introducing an objective function that penalizes certain types of solutions. In other words, we seek a solution that is consistent with our a priori knowledge of what a "good" solution should look like by penalizing solutions that deviate from this expectation. For example, we could attempt to solve

$$\min \|\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{\Psi x}\|_2^2 \leq \epsilon \tag{6}$$

where the second term expresses our desire for a solution that is consistent with the observed data and the first term penalizes large coefficients (in this case the goal is to find a minimum energy representation). When $\epsilon = 0$ the solution to this set of equations corresponds to the pseudo-inverse. Unfortunately, the pseudo-inverse tends to result in solutions which contain many small non-zero coefficients (and this tends to complicate the interpretation and use of the additive model) [14].

Another approach to counteract the ill-posedness of the interpolation problem is to penalize non-sparse solutions. There are many reasons for preferring sparse solutions as highlighted at the end of Sec. 2. We may leverage the fact that musical signals tend to be relatively sparse when represented using Gabor atoms and the additive model. For example, it is well-known that sparse signals can be reconstructed using a limited set of measurements (this is the basis of compressive sensing) [9].

In Adler et al. [5] sparse approximation was suggested as a tool to regularize the audio interpolation process. In particular, the orthogonal matching pursuit (OMP) algorithm was used to estimate the additive model coefficients in Eq. (5).

An alternative way to find sparse representations is to replace the 2-norm in Eq. (6) with the 1-norm which leads to the basis pursuit denoising (BPDN) optimization problem [15]:

$$\min \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{\Psi}\mathbf{x}\|_2^2 \leq \epsilon \tag{7}$$

The 1-norm is attractive seeing as it is convex (so convergence to a local minimizer is guaranteed) and because it often induces sparse solutions [16].

In [7] it was shown that additional structure exists between the non-zero coefficients in sparse atomic models of audio. Specifically, it was shown that time-frequency representations of audio tend to exhibit a high degree of continuity between temporally adjacent atoms. This result is due to the fact that musical sounds tend to be somewhat stable (e.g., the decay time of a resonant mode tends to be longer than the length of individual atoms, and therefore multiple adjacent atoms tend to be activated simultaneously). In the following section we describe how recent results on structured-sparse modelling of audio can be applied to the interpolation problem. We begin by reviewing the proposed optimization problem and then discuss algorithms for its solution (a more detailed treatment of this formulation can be found in [7]).

4.2 Structured-sparse estimation

We would like to modify the BPDN optimization problem in order to exploit joint relationships between the representation coefficients. To this end we propose the following generalization of the BPDN problem (termed G-BPDN) for structured-sparse estimation:

$$\min \|f(\mathbf{x})\|_1 \quad \text{subject to } \|\mathbf{z} - \mathbf{\Psi}\mathbf{x}\|_2^2 \leq \epsilon \tag{8}$$

Ideally, the function f should sparsify $\mathbf{x}$. This in turn allows us to use a BPDN-like formulation with coefficient vectors that are sparse after some transformation. This formulation is similar to the co-sparse analysis formulation from [17], however, we do not restrict f to be a linear operator. For the task of audio interpolation we propose using then following G-BPDN objective function:

$$\|f(\mathbf{x})\|_1 = \|\mathbf{L}|\mathbf{x}|\|_1 \tag{9}$$

where $|\mathbf{x}| = [|x_1|, |x_2|, \ldots, |x_n|]^{\mathrm{T}}$ is a vector containing the magnitudes of $\mathbf{x}$ and

$$\mathbf{L} = \begin{bmatrix} \tilde{\gamma}\mathbf{D} \\ \gamma\mathbf{I} \end{bmatrix} \tag{10}$$

In our case the analysis operator $\mathbf{D}$ is a matrix designed to calculate the amplitude difference between temporally adjacent pairs of coefficients and $\tilde{\gamma} = 1 - \gamma$. The parameter $\gamma \in [0, 1]$ can be used to emphasize either sparsity or amplitude continuity, however, in this work we simply fix $\gamma = 0.5$ (in which case it drops out of the optimization problem).

In this form, the G-BPDN optimization problem can be used to emphasize sparsity of the time-frequency coefficients as well as the sparsity of their time derivative (which should produce solutions with greater temporal continuity). It should be noted that the proposed optimization problem is quite similar to both the fused-lasso [18] and total-variation denoising [19] which are well-known in the statistics and image processing literature, respectively.

We propose solving this problem via smoothed projected gradient descent as outlined in [7]. We note that other techniques, such as the alternating direction method of multipliers (ADMM), could be used as well [20]. Projected gradient descent is a two-step procedure: a gradient descent step is taken and the result is projected onto the set of feasible solutions [21, 22]. The steps of this algorithm are outlined in Alg. 1.

Algorithm 1 Projected Gradient G-BPDN

1: **init:** $\mathbf{x}^{(0)} = \mathbf{\Phi}^H\mathbf{z}, n = 0$
2: **repeat**
3: $\quad \mathbf{u}^{(n)} = \mathbf{x}^{(n)} - \mu \cdot \text{diag}(S_\infty(\mathbf{x}/e))\mathbf{L}^T\mathcal{S}_\infty(\mathbf{L}|\mathbf{x}|/e)$
4: $\quad \lambda^{(n)} = \max\left(0, \epsilon^{-1/2}\|\mathbf{z} - \mathbf{\Psi}\mathbf{u}^{(n)}\|_2 - 1\right)$
5: $\quad \mathbf{x}^{(n+1)} = \mathbf{u}^{(n)} + \frac{\lambda^{(n)}}{1+\lambda^{(n)}}\mathbf{\Psi}^H(\mathbf{z} - \mathbf{\Psi}\mathbf{u}^{(n)})$
6: $\quad n = n + 1$
7: **until** stopping condition

In Alg. 1, line 3 corresponds to the gradient descent step and line 5 corresponds to a projection onto the feasible set [1]. The operator $\mathcal{S}_\infty$ denotes projection onto the inf-norm ball and e is a smoothing parameter (as outlined in [7]). In general we stop this algorithm when the change in the objective function from one iteration to the next is small (e.g., less than 10^{-6}), however, a maximum number of iterations may be enforced as well.

5. RESULTS

5.1 Interpolation results

In this section we provide results that demonstrate the efficacy of our structured-sparse estimation approach. We focus specifically on an experiment outlined in Adler et al. [5] so that we may directly compare our results against a recent (and similar) approach. The test data consist of 10, five seconds music signals sampled at 16kHz which are available from [5]. Each of these audio excerpts was corrupted periodically (every 100ms) by setting an interval of samples to zero. The size of the missing interval duration was varied from a fraction of a millisecond up to 10ms in order to gauge how the performance would change with respect to the amount of missing data. A mask matrix was manually created to identify the missing samples. In the

[1] Provided $\mathbf{\Psi}$ is a Parseval tight frame as outlined in [7].

following experiments we used a tight frame Gabor dictionary created using Hann windows of length 64ms with 75% time overlap [2]. Furthermore, we set the parameter $\epsilon = 10^{-10}$ in order to force our model to represent the known samples with virtually no error [3].

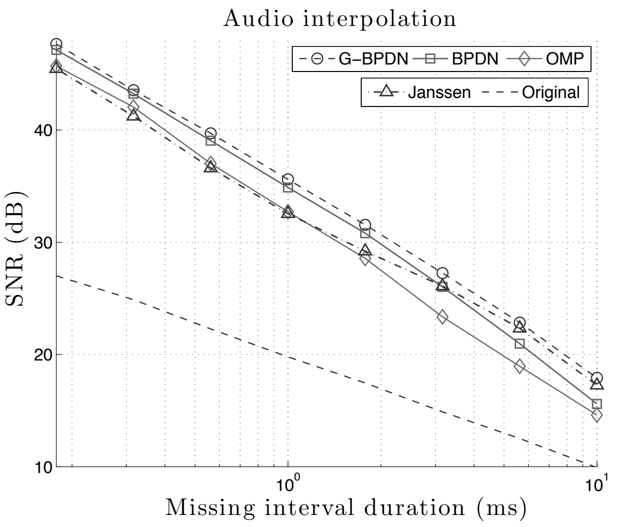

Figure 1. Results of interpolation for missing intervals of various durations. Results averaged over 10 test signals.

The interpolation performance was evaluated by measuring the signal-to-noise ratio (SNR) between the true signal and the residual error:

$$\text{SNR} = 10 \log_{10} \frac{\|\mathbf{y}\|_2^2}{\|\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}\|_2^2} \tag{11}$$

where $\hat{\mathbf{x}}$ is the vector of estimated model coefficients.

Fig. 1 illustrates the SNR vs. missing interval duration averaged over all 10 test signals for both sparse and structured sparse interpolation (e.g., the solutions to Eq. (7) and Eq. (8), respectively). This graph also shows the results obtained using code from [5] (labelled as OMP) and the AR model from [1] (labelled as Janssen). The G-BPDN interpolation obtains the highest SNR in all cases. The benefit of G-BPDN (over the purely sparse estimation) is also more readily apparent as the missing interval duration grows. This illustrates that solutions with greater temporal continuity are indeed beneficial for bridging larger gaps of missing samples. The AR model from Janssen [1] also performs well for large gaps and even outperforms the purely sparse solution in this case. This is presumably because the AR model contains a memory of the previous samples (and therefore better models the temporal structure of the signal).

Fig. 2 illustrates the interpolated waveforms for a missing interval of 10ms (for a single test sound). It is evident that the G-BPDN solution is slightly closer to the true waveform than the other estimates. However, one can not automatically conclude that an improvement will be perceived when listening to the interpolated sounds. Indeed, when listening to sounds interpolated with BPDN and G-BPDN the results are very similar. This may, however, be a consequence of the fact that the auditory system is capable of filling in short gaps in missing sounds, which is a well-known fact (see for example, [23]). We have included several examples of interpolated sounds on the companion website [24].

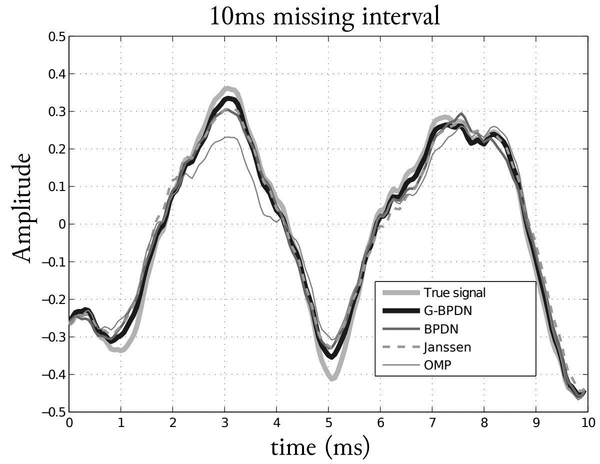

Figure 2. Waveform plot showing the interpolated results for a single test signal over a missing interval of 10ms.

From a computational point-of-view BPDN and G-BPDN have approximately the same complexity (each is dominated by two matrix vector products with the dictionary at each iteration). The overall complexity will depend on the number of iterations required for convergence which is difficult to predict a priori. A possible advantage of our approach is that after each iteration the coefficients $\mathbf{x}$ will satisfy $\|\mathbf{z} - \mathbf{\Psi}\mathbf{x}\|_2^2 \leq \epsilon$. In other words, the representation error is always bounded. An analysis of the computational complexity of OMP can be found in [25]. Quite informally we note that BPDN and G-BPDN both ran much faster (between 5-10$\times$ faster for a full run) than OMP on the same computer (all algorithms were implemented in Matlab and run on the same data).

5.2 Morphing results

As mentioned in Sec. 3.2 we can use the interpolation framework in order to transition between a source and target sound. To recap: we simply include a gap of missing samples between the two sounds which is subsequently treated as missing data to be interpolated. In general, the amount of data we want to estimate is much greater when transitioning between two sounds (in comparison to typical restoration tasks). The atoms in the Gabor dictionary should, at minimum, span the interval we wish to interpolate. As a rule of thumb we have found that atoms anywhere from 2 to 4$\times$ the gap length produce good results. As this leads to very long atoms in practice, it is wise to ensure that the number of known samples is at least this large as well.

[2] In this work the set of dictionary atoms span the space of the input signal. Block/frame-based processing was not used.

[3] We note that the value of ϵ we use is smaller than the one used in [5]. It is difficult to say whether or not this effects the results since the algorithms used are completely different. The value of ϵ used in this work was optimized to obtain the best result for the algorithm we considered (and we assume the authors in [5] would have done the same as well).

When estimating the additive model coefficients using Eq. (8) we have some flexibility with regard to the parameter ϵ, which controls the degree of approximation error that we are willing to tolerate (with respect to the known samples). As we increase the value of ϵ the set of feasible solutions grows, which in turn means that solutions with a greater degree of temporal continuity may be found (although these solutions will no longer perfectly match the known data). This flexibility can be beneficial when transitioning between two sounds since we often want the transition to be as smooth as possible. We have also found that novel sounds/timbres can be created with large values of ϵ (these can be heard online as discussed below).

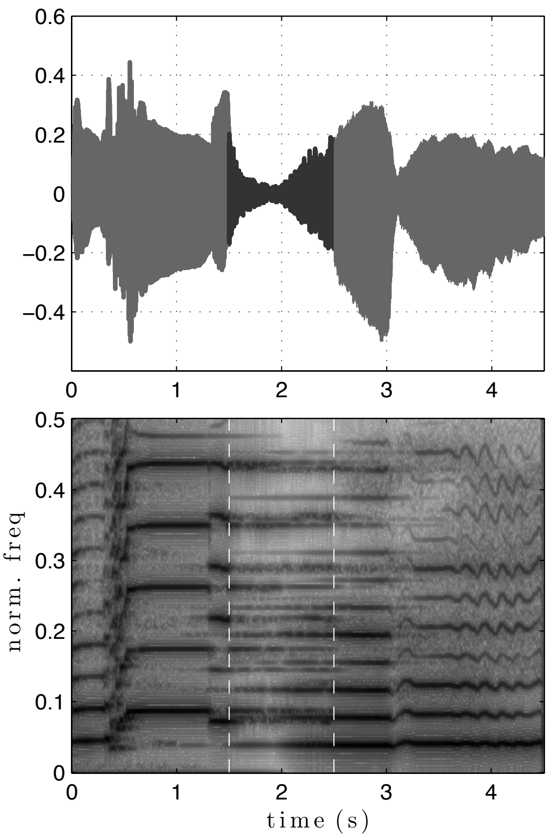

Figure 3. A one second transition between a saxophone passage and a female vocal excerpt (the sampling rate was 16kHz). Top: known data (light gray), interpolated data (dark gray). Bottom: spectrogram.

Fig. 3 shows a one second transition between a saxophone passage and a female vocal excerpt. For this example a Gabor dictionary with 2s long atoms was used and ϵ was set to 10^{-10}. The spectrogram seems to indicate an extension and averaging of the partials from each sound in the transition region. The audible impression for this particular transition is that the start/end of each sound has been extended into the transition region by adding reverb and preverb (i.e., reverb that precedes the sound).

We cannot quantitatively assess the morphing results (since no ground truth data exists for such a task). In order to qualitatively test the morphing results we have experimented with a wide variety of sources/targets including relatively stable sounds (e.g., clarinet, trumpet, tuba) and sound textures (drums, noise, abstract sounds). We also tested several missing interval durations from very short durations (250 ms) to extremely long durations (3 s). We have posted several audio examples on the companion website [24].

We make the following qualitative observations regarding these sounds. Firstly, for stable smooth sounds the morphing is quite similar to a simple cross-fade, although our transition appears to be slightly smoother. However, for more complicated sounds the results are quite audibly different from a simple cross-fade (the timbre during the transition appears to be more of a hybrid than a simple sum of the two signals). As mentioned above the effect sometimes sounds as though the source and target have been extended into the transition region by adding reverb and preverb. Indeed, since the atoms used are very long, their tails extend into the transition region, which helps to create this effect.

In our examples the cross-faded sounds are somewhat shorter than the interpolated sounds. This is because we must overlap the source and target sounds when we make a cross-fade. This could certainly be remedied by using more data for the cross-fade, however it brings to light a benefit of our interpolation approach: since we generate entirely new data for the transition, we can create longer transitions using less source material. This might be valuable in certain situations where the amount of available data is limited (for example, transitioning between tiny slices of sound which is common in some genres, e.g., 'microhouse').

We also note that tuning the value for ϵ allows us to create a wide variety of different sounds (some of which sound more 'wet' and others which sound more 'dry'). The ability to tune ϵ is a major advantage of our technique since it leads to many interesting transition effects.

6. CONCLUSION

We have presented a method of audio interpolation that can be used to restore missing or corrupted audio data. We began by modelling the sound as a linear combination of time-frequency atoms. Then, based on the observation that many musical signals are simultaneously sparse and structured (in terms of temporal continuity between the additive model coefficients), we proposed a structured-sparse optimization problem for estimating the model parameters. This model was subsequently used to synthesize an estimate of the missing samples. We compared our strategy to several state-of-the-art interpolation schemes and showed that, on average, our approach leads to an improvement in terms of the SNR. We also highlighted how this process can be used to morph/transition between sounds and provided several audio examples representative of the kind of results which may be achieved. Future work will examine additional types of structure/constraints that can be leveraged to improve the interpolation procedure. Finally, it would be interesting to consider a non-local approach to interpolation, especially for signals that are highly nonstationary. For example, one could try to integrating the ideas in [26] within the sparse approximation framework.

Acknowledgments

The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

7. REFERENCES

[1] A. Janssen, R. Veldhuis, and L. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 34, no. 2, pp. 317–330, 1986.

[2] W. Etter, "Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters," *IEEE Transactions on Signal processing*, vol. 44, no. 5, pp. 1124–1135, 1996.

[3] S. Godsill, P. Rayner, and O. Cappé, *Digital audio restoration*. Springer, 2002.

[4] M. Lagrange, S. Marchand, and J. Rault, "Long interpolation of audio signals using linear prediction in sinusoidal modeling." *J. Audio Eng. Soc.*, vol. 53, pp. 891–905, 2005.

[5] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. Plumbley, "Audio inpainting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 922–932, 2012.

[6] M. Elad, J. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340–358, 2005.

[7] C. Kereliuk, "Sparse and structured atomic modelling of audio," Ph.D. dissertation, McGill University, defended on March 28th, 2013.

[8] C. Kereliuk and P. Depalle, "Sparse atomic modeling of audio: A review," *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pp. 81–92, 2011.

[9] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] F. O'Reilly Regueiro, "Evaluation of interpolation strategies for the morphing of musical sound objects," Master's thesis, McGill University, 2010.

[11] K. Fitz, L. Haken, S. Lefvert, and M. O'Donnel, "Sound morphing using LORIS and the reassigned bandwdith-enhanced additive sound model: Practice and applications," in *Proc. International Computer Music Conference*, 2002, pp. 393–400.

[12] A. Olivero, B. Torrésani, P. Depalle, and R. Kronland-Martinet, "Sound morphing strategies based on alterations of time-frequency representations by Gabor multipliers," in *Proceedings AES 45th International Conference on Applications of Time-Frequency Processing in Audio*, 2012.

[13] M. Caetano and X. Rodet, "Sound morphing by feature interpolation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 161–164.

[14] M. Goodwin, "Matching pursuit with damped sinusoids," in *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 1997.

[15] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[16] D. Donoho, "For most large underdetermined systems of equations, the minimal ℓ_1-norm solution approximates the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 907–934, 2006.

[17] S. Nam, M. Davies, M. Elad, and R. Gribonval, "Cosparse analysis modeling-uniqueness and algorithms," in *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5804–5807.

[18] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[19] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, pp. 259–268, 1992.

[20] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2345–2356, 2010.

[21] S. Becker, J. Bobin, and E. Candès, "NESTA: a fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[22] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[23] R. M. Warren *et al.*, "Perceptual restoration of missing speech sounds," *Science*, vol. 167, no. 3917, pp. 392–393, 1970.

[24] C. Kereliuk. (2013). [Online]. Available: http://www.music.mcgill.ca/~corey/smc2013

[25] B. Mailhé, R. Gribonval, P. Vandergheynst, and F. Bimbot, “Fast orthogonal sparse approximation algorithms over local dictionaries,” *Signal Processing*, vol. 91, no. 12, pp. 2822–2835, 2011.

[26] M. Niediwiecki and K. Cisowski, “Smart copying-a new approach to reconstruction of audio signals,” *Signal Processing, IEEE Transactions on*, vol. 49, no. 10, pp. 2272–2282, 2001.