IRP: Fourier priors for recovering audio data from damaged signals

Mair Allen-Williams (s0454371)
Supervisor: Amos Storkey

1 Introduction

Quantities of archived audio data exist in numerous contexts; examples include radio recordings, recordings of concerts, home-made tapes. Some of these may have been poor quality recordings, others may have degraded over time or been damaged at some point. We therefore consider techniques for improving the quality of audio data. Damaged data may be noisy, clipped, poorly compressed, or have gaps or clicks in the stream from damaged media.

There are some existing methods for doing this; CD players commonly attempt resolution enhancement; most audio editing tools include 'noise removal' functions of some quality. Various companies use proprietary techniques to combine approaches, resulting in effective noise removal. There is a wealth of current research into audio restoration techniques, using techniques which range from simple interpolations through statistical models to neural networks and psychoacoustic analysis.

The nature of audio data is such that spectral (frequency-domain) approaches are often appropriate. I propose to extend existing techniques to form a system based on the fast Fourier transform; the graphical data structure which facilitates efficient algorithms. By using conditional spectral priors based on earlier states I hope to make use of temporal correlations to improve on existing spectral techniques.

Archived data, particularly in formats such as audiotape, may suffer from decay over time. There is therefore a clear application for an effective and efficient way of restoring large quantities of archived data. At the same time, growing storage capacities and reducing costs of technology mean that more than ever home recordings are being made and saved—possibly low quality, or in noisy conditions, such as a video recording taken on a windy day by a cheap digital camera. Widespread use of the internet has created a niche for compressed audio and low bandwidth transmission.

As well as practical applications, there will also be research benefits: the efficient noisy FFT—which can be used for a wide variety of applications besides audio—, and the use of good prior representations for audio data and error models. The techniques used in the project will be extensible to other domains, such as image restoration. The JPEG algorithm performs a cosine transform on quantised datapoints. Using spectral priors we can more accurately estimate the original datapoints. Another audio application is source separation, estimating the original signals from a mixed signal. Fast and effective methods for this problem are still lacking, as increasingly effective techniques demand increasingly high computational resources. Current techniques are typically limited to perfect sources. Assuming perfect sources when the truth is noisy results in separation into noisy sources, while ideally we would like to remove the noise as well as separating the signals. Beyond audio applications, there are many other domains in which a Fourier transform with noisy or missing data can be applied: astrophysical data, image data, nuclear magnetic resonance signals, and a range of other periodically varying signals.

2 Prior work

2.1 Audio restoration

Bayesian models Godsill and Rayner [10] give an overview of Bayesian model-based approaches to removal of clicks (in the time domain), hiss (in the frequency domain) and other defects. These include the popular Kalman filter [21]—based on a prediction of later states from current states, Monte Carlo methods such as the Gibbs sampler, the Wiener filter for hiss reduction, and more sophisticated methods involving the use of E-M in the interpolation of missing data. Fong and Godsill [7] describe a way of exploiting substructure in Monte Carlo techniques to improve both efficiency and performance. Troughton [20] uses Bayesian models with Monte Carlo sampling to restore quantised data. Godsill and Davy [6] describe how to incorporate harmonics into a Bayesian model, and suggest that future audio restoration work could make use of this information.

These statistical models give good results, but do not handle non-linear distortion, for example clipping. Godsill et al [9] touch on these in their generalised overview statistical model-based approaches, describing the relevant models. Although their experimental results on clipped and quantised data show some improvement in the audio signal, they describe the computational requirements as 'prohibitive' and observe that they have had to make several simplifying assumptions. Time-domain interpolations are also not suited to filling in longer blocks of missing data. [10] mentions an interpolation which operates in the frequency domain and is suitable for filling in longer blocks.

Neural networks Czyzewski [5] describes an approach to removing impulse noise based on learning a neural network. His approach uses two networks: one to detect the disturbances, and one to recover the original data. The networks are trained on both clean and distorted data. The disadvantage of his approach is that training neural networks is a time-consuming process, and distinct networks must be trained—requiring a good body of training material—for distinct classes of either audio signal or impulses. Czyzewski develops the use of neural networks further in [4], using the self-organising map of Kohonen and a neuro-rough controller (one which uses *rough* sets [13]—that is, creating approximations of sets) to remove non-stationary noise. The computational resources required for training remain an issue. Cocchi and Uncini [3] describe a neural network approach, operating in the frequency domain, to interpolating large blocks of missing data. They emphasize the advantages of non-linearity that neural networks provide. Their use of subband rather than full-band analysis ameliorates the performance difficulties somewhat.

2.2 Spectral approaches to audio data

As well as the spectral methods touched on above, there are a number of other spectral techniques for manipulating audio data. data. Simple examples are filters which remove unwanted frequencies; hiss removal is usually performed in the frequency domain. Scott and Wilson [16] use a multiresolution Fourier transform to restore audio signals. Their approach relies on the existence of a target or *prototype* signal, possibly employing (for example) musicians to give a rendering of a recorded piece which is being restored.

Away from audio restoration research, there is work on using spectral techniques to improve audio analysis. This work could be incorporated into audio restoration techniques. For example, Cai et al [2] use various feature structure patterns to distinguish different types of audio signal. They also mention the use of principal component analysis in the frequency domain for de-noising.

Roweis [15] describes an approach to the source separation problem using a factorial hidden Markov model on spectrograms (a spectrogram is a representation constructed from the Fourier transform). Bach and Jordan [1] also use a spectrogram-based technique to solve the same problem. However, their methods are computationally heavy and resource intensive, involving large matrix calculations.

Operating in the frequency domain therefore looks like a promising approach. However, as well as the issues of efficiency mentioned, the algorithms referenced above cannot handle missing data. There are algorithms such as the time-series interpolation algorithms mentioned previously which could estimate the missing data. However, errors have a tendency to compound; it is likely to be advantageous to manage missing data as part of a unified model.

There has been some work on using a Bayesian derivation of the Fourier transform to estimate the transform of incomplete data. Gregory [12] describes a method due to Jaynes [14], using both systems with strong prior information and systems where the only prior information is in the choice of model. Another interesting approach is that of Storkey [19]: this method is designed to solve the efficiency problems mentioned above, and handle noisy data as well as missing data.

Storkey's technique is based on incorporating the model directly into the Fourier transform, exploiting the graphical structure of the efficient fast Fourier transform algorithm. The work is tested on missing data only, but should also be applicable to noisy data of other kinds.

2.3 Conclusions

There are two common ways of handling audio data: in the time-domain, looking at the time series of the data, and in the frequency domain, using the Fourier transform of the data. The two approaches have strengths for different kinds of error. Current state of the art techniques generally use a series of probabilistic techniques each operating on either the time-series or the frequency transform, aimed at removing different kinds of noise. Good models for handling non-linear distortions, and estimating longer blocks of missing data are still needed: some work has been done into neural network approaches, but this have strong computational disadvantages.

Spectral approaches are a promising way of handling various types of audio restoration, in particular clipping and large gaps which are not handled by the current time-series approaches. Beyond spectral models specifically aimed at interpolation of missing data, current spectral techniques for handling audio do not deal well with the absence of datapoints, and some struggle with inefficiency. Storkey [19] proposes a way of incorporating the noise model directly into the Fourier transform, making use of the structure of the efficient fast Fourier transform algorithm. It seems likely that this will provide an efficient and flexible way of handling noisy audio data.

2.4 Extensions

Source separation A mixed signal can be viewed as a single signal with added noise. This single signal could be extracted from data using the noise removal approach. This extraction could be repeated on the remaining data until the correct number of audio signals had been collected, leaving only noise. The effectiveness of such an approach would depend on the quality of the model; in particular the use of conditional priors will be important in identifying a single signal from several plausible ones.

Image data The above FFT approach can also be applied to images. Storkey and Allan [18] have done worked on reconstructing compressed JPEG images using spectral priors for the cosine transform. The work could be extended to use conditional priors. Another extension would exploit the fact that image data is inherently two-dimensional, and looking for patterns in the data as a single string of datapoints is sub-optimal. The belief propagation technique described above could be extended to multi-dimensional versions of the Fourier transform.

3 Methods

3.1 Data sources

I will need three forms of audio data. The first, which is more readily available, will be damaged archive data, in the forms described above. Examples used in the work of Wolfe([22],[9]) and Godsill([10],[11],[7]) are available from their websites (`http://people.deas.harvard.edu/~patrick/research/`, `http://www-sigproc.eng.cam.ac.uk/~sjg/springer/`, `http://www-sigproc.eng.cam.ac.uk/~sjg/clipping/`). The second, which enables more accurate quantitative testing, is undamaged data. While perfect data may not exist, high quality audio files can be taken from CDs. These can be damaged in artificial, controlled ways. Finally, I will use synthetic data—manually created examples of simple features which we wish to explore, and more complex examples.

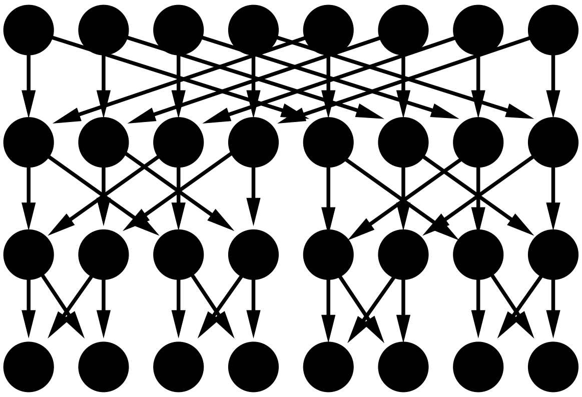

Figure 1: Fourier transform as belief network

3.2 Models

I intend to explore approaches using different kinds of model for the spectral priors and different kinds of error model on the data. It will therefore be necessary to write down a set of models which encode either simple or more involved assumptions about both the clean data and the errors in the data. The models should also handle the cases where more than one type of error has occurred, either in serial or parallel.

3.3 Core: Noisy FFT implementation

The fast Fourier transform can be viewed as a belief network (figure 1). Values are assigned probabilistically to nodes and belief propagation techniques are used to estimate the probability of values at unknown nodes.

Belief propagation The chief issue in the FFT implementation is the method of belief propagation. The network defined by the FFT is not singly connected which means that we cannot use exact propagation methods. Storkey [19] used loopy propagation. However, this has not been fully explored and suffers from instabilities in certain cases. For simplicity I propose to use a preconditioned conjugate gradient approach. This is an exact optimisation technique which is known to result in good approximations in reasonable time. Conjugate gradient methods [17] are widely used and there are available implementations of both conjugate gradient methods and preconditioners which I can use. Before starting work on the project I will need to become familiar with these techniques.

3.4 Baseline system

After writing and testing the core code to implement inference in a FFT network, I will write code to accept audio data and uses the noisy FFT to try and improve the audio quality using simple independent Gaussian priors on the spectra.

3.5 Extended system

I will then extend the baseline system, relaxing the assumptions of independence on the spectra, making the priors conditional on the previous state (this has some parallels with Roweis' source separation work using HMMs [15]). Next, the assumption of Gaussianity can be relaxed. The choice of directions at this take will be suggested by the evaluation of earlier results; following the typical cycle of using an error analysis to decide on the next step to take on improving the system. It will be necessary to ensure a broad range of coverage of types of data, types of damage, and choice of models.

3.6 Outputs

The primary output of the system will be the modified audio data. I will also record analyses of how the files were modified and the extent of the modifications.

3.7 Design

The core noisy Fourier transform code is time-critical; this is likely to be the bottleneck in the system. I therefore propose to use C for this part of the project. For the rest of the code I propose to use Matlab, which will allow simple and compact representations of the data and mathematical operations. C and Matlab are straightforwardly interoperable. I will need to design the Matlab system so that audio files can be partially read in and computations performed on 'windows' of data; memory requirements will otherwise become an issue for large files.

Good software engineering practice will be vital in keeping to timetables. The project is suited to the waterfall method of development; fully designing, building and testing each stage before proceeding to the next stage.

4 Evaluation

Given an imperfect or damaged representation of an audio signal, we will attempt to reconstruct the original signal. Evaluation takes the form of measuring how close our reconstruction is to the original, compared with the damaged signal. I also proposed that we could perform this transform efficiently. It will therefore be necessary to measure the performance of the system. Finally, the results should be compared with those produced by existing methods.

There is no need to use complicated evaluation methods. I will define a small number of simple measures and heuristics, which coupled with the qualitative measuring should give a fair estimate of how well the system is doing. Canazza et al [8] give an overview of some audio evaluation methods, using factor analysis with subjective testing to determine which errors are most disturbing to humans.

4.1 Undamaged and synthetic data

If there is an authoritative form of the data, then we can define evaluation measures which compare the 'repaired' data with the undamaged data. For example, we could simply compare the values of the datapoints, both in time-space and in Fourier space, and compute the mean-squared error. Another possible measure is the signal-to-noise ratio.

The focus of audio restoration is on making a signal clearer to humans, and our error estimates should reflect this, for example by penalising errors more heavily when they are compounded close together. An isolated erroneous datapoint among accurate datapoint would hardly be detectable and barely needs penalising. In Fourier space, measures can also take into account that some frequencies are less detectable by humans than others. Wolfe and Godsill [22] employ a cost function to incorporate perceptual information into their model. Similarly, cost functions can be incorporated into the error model.

We can also compare Fourier space features in the undamaged and the 'repaired' data. Roweis [15] describes continuity features, common fate features (elements such as offset/onset with common time variation), and speech related features such as pitch estimation. Harmonic frequencies should travel along with their major chord.

Finally, the heuristic and qualitative evaluation measures used below will be vital in sanity-checking and supplementing these measures.

4.2 Damaged data

Heuristics to estimate ‘goodness’ of audio samples will include features similar to those described above. Qualitative testing along with examination of the data can be used to suggest other heuristics, using the undamaged data to test them. I will aim for a broad coverage with these heuristics; it should be the case that if an audio file scores highly on this heuristic testing it does actually sound good.

All tests should also compare the repaired data with the original damaged data; it’s no good having a wonderful rendition of the Moonlight Sonata if the original file was the Spice Girls’ Wannabe. In general we can assume that most of the points in the original file will not vary much.

Perhaps the most important evaluation measure in an application to make audio data ‘sound’ better will be qualitative testing: listening to the results and determining ‘how they sound’. Several second opinions will be useful; I will enlist other humans, asking them to scale how ’good’ the cleaned audio sounds, and average the results. It will also be important to get specific comments on which of the remaining errors are more noticeable. As well as being informative in itself, this kind of human-based testing alongside the above testing will help justify the choice and implementation of quantitative measures.

4.3 Models

The outputs can only be as good as the models. Part of the evaluation will report which models are more effective on which kinds of data. This will mean evaluating the results of different models on the data. For the synthetically damaged data, the correct model will be known; for the rest we will have to estimate.

4.4 Efficiency

Performance measures include cpu usage, running time, memory usage. While I will comment on memory usage, I will focus on speed. I will consider the running time for variously sized files and error models on a fixed machine, and comment on the feasibility and the comparison with previous tests (taking into account the specifications of the relevant machines).

4.5 Comparison with other methods

In order to determine whether the system is practically useful or not, we should compare it with other solutions of the same problem. The samples available on the websites of Springer and Wolfe include the restored versions. I can compare with these, commenting on both speed and effectiveness. The broad coverage of the evaluation measures is vital here, and it will be important to use the qualitative, human-based, testing for these comparisons to justify any claims of ‘better’ performance suggested by the quantitative evaluation. I will also compare with published results in the papers I have described above.

5 Further work

Two extensions to the system could be made if the project moves along unexpectedly fast. It is unlikely that there will be time to implement these extensions.

We can use the core system to perform noisy source separation, extracting one signal at a time. Similar sets of priors can be used. The same set of evaluation metrics are appropriate, evaluating each output against each input.

The core system could also be used to accept image data. Similar Matlab code to that which reads in the audio data would read in image files and pass it to the FFT code. The choice of priors would need to be modified to model image data, although the baseline system of independent Gaussian priors would be much the same. An extended version of the core, handling two-dimensional Fourier transform could exploit the two-dimensional nature of images. These image restoration systems would use a similar set of evaluation metrics, but a different set of heuristics.

6 Resource requirements

I will need sufficient storage for a comprehensive collection of audio data. In order to process large numbers of audio files it will also be necessary to make use of considerable cpu and memory resources. I will have access to several clusters of machines in the DICE system. I will also need several people to perform qualitative testing.

7 Deliverables

From the above, the following list of deliverables can be extracted. I do not reiterate the details.

- Tested noisy FFT core code.
- Tested baseline system for improving audio data.
- Tested improved system for improving audio data, using non-Gaussian and conditional spectral priors.
- A report on the work (dissertation), including an evaluation of the system on different types of damaged data.

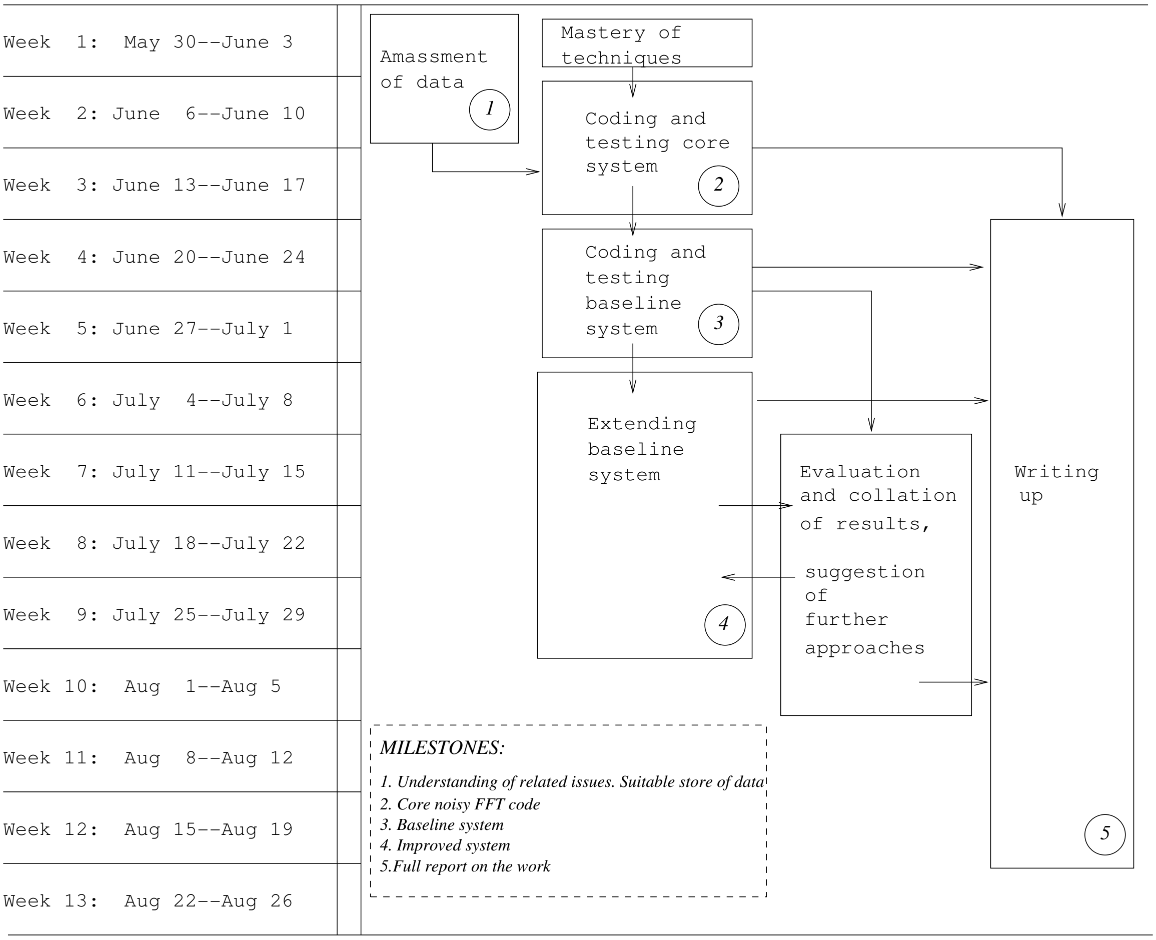

Figure 2: Work plan

8 Workplan

Figure 2 shows the timetable for the work, with milestones corresponding to the preliminary work and deliverables marked on it. The waterfall nature of the bulk of the system can be clearly seen, with the writeup running alongside. We allow some flexibility in the final module in the system, using feedback and error analyses from the earlier results to suggest directions for later work to take. I expect to be continually improving both the system and the error models at this stage.

References

[1] Francis R. Bach and Michael J. Jordan. Blind one-microphone speech separation: A spectral learning approach, 2004.

[2] Rui Cai, Lie Lu, Hong-Jiang Zhang, and Lian-Hong Cai. Improve audio representation by using feature structure patterns. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004.

[3] G. Cocchi and A Uncini. Subband neural networks prediction for on-line audio signal recovery, 2002.

[4] Andrzej Czyewski and Rafal Królikowski. Neuro-rough control of masking thresholds for audio signal enhancement. *Neurocomputing*, 36(1-4):5–27, 2001.

[5] Andrzej Czyzewski. Learning algorithms for audio signal enhancement: Part 1 neural network implementation for the removal of impulse distortions, 1997.

[6] M. Davy. Bayesian harmonic models for musical pitch estimation and analysis. Technical Report CUED/F-INFENG/TR.431, Engineering Department, University of Cambridge, UK, April 2002.

[7] W. Fong, S. Godsill, A. Doucet, and M. West. Monte carlo smoothing with application to audio signal enhancement.

[8] Canazza S.; Coraddu G. and De Poli G.; Mian G.A. Objective and subjective comparison of audio restoration methods, 2001.

[9] S. J. Godsill, P. J. Wolfe, and W. N. W. Fong. Statistical model-based approaches to audio restoration and analysis., 2001.

[10] Simon J Godsill and Peter J W Rayner. Springer-Verlag, 1998.

[11] S.J. Godsill and P.J.W. Rayner. Robust noise modelling with application to audio restoration, 1995.

[12] P.C. Gregory. A Bayesian revolution in spectral analysis. 1997.

[13] http://www2.cs.uregina.ca/ roughset/. Electronic bulletin of the rough set community.

[14] E. Jaynes. Bayesian spectrum and chirp analysis, 1987.

[15] Sam T. Roweis. One microphone source separation. In *NIPS*, pages 793–799, 2000.

[16] H.R.R. Scott and R. Wilson. A multiresolution audio restoration algorithm. pages 151–154, 1995.

[17] Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, 1994.

[18] Amos Storkey and Michael Allan. Cosine transform priors for enhanced decoding of compressed images, 2004.

[19] Amos J. Storkey. Generalised propagation for fast fourier transforms with partial or missing data. In *NIPS*, 2003.

[20] Paul T. Troughton. Bayesian restoration of quantised audio signals using a sinusoidal model with autoregressive residuals, 1999.

[21] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, 1995.

[22] Patrick Wolfe and Simon Godsill. Perceptually motivated approaches to music restoration.