Scatter: A Software Tool for Visualizing, Transforming, and Performing Atomic Decompositions of Sound

[Extended Abstract]

Aaron McLeran
Media Arts and Technology Program
University of California
Santa Barbara, CA
93106-9560
amcleran@mat.ucsb.edu

Curtis Roads
Media Arts and Technology Program
University of California
Santa Barbara, CA
93106-9560
clang@mat.ucsb.edu

Bob Sturm
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA
93106-9560
boblsturm@gmail.com

ABSTRACT

This paper presents a software tool which allows a user to visualize and transform atomic decompositions of sound produced by matching pursuit. The matching pursuit algorithm, which is an atomic, dictionary-based signal decomposition method, can be seen as the analytical counterpart to granular synthesis. Matching pursuit iteratively matches a given signal with grains, or atoms, from a user-defined dictionary (or set of parameter definitions) of basis waveforms. When the algorithm is finished, we are left with a granular model, or atomic decomposition, of the original signal which is called a book.

Scatter provides a GUI interface for performing the matching pursuit analysis and shows the decomposition build as the algorithm progresses. Scatter then displays the Wigner-Ville distribution and energy contribution of each atom in the decomposition in three dimensions. The third dimension represents the spatial position of each atom.

Scatter also gives the user multiple ways to transform the decomposition: by direct interaction with mouse selection, by parametric filtering, and by application of stochastic algorithms. Scatter also provides a means for real-time performance through the use of a variety of decomposition scrubbing techniques. Scrubbing is a process that allows a user to synthesize any desired region of the decomposition.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Methodologies and techniques, Signal analysis, synthesis, and processing* ; D.2.6 [**Software Engineering**]: Programming Environments —*Graphical environments, Interactive environments*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI), Interactive environments*

General Terms

EXPERIMENTATION, DESIGN

Keywords

Sound Analysis and Synthesis, Media Signal Processing, Electronic Music, Software

1. INTRODUCTION

Methods for deriving atomic decompositions using dictionary-based methods (DBMs) have been developed in the domain of signal processing engineering with the intention of finding sparse ways to represent signals which are robust to noise, meaningful with respect to the original signal, and malleable to transformations [3, 1].

DBMs give a user the ability to choose the basis waveforms (or functions) for a particular dictionary. This allows dictionaries to be adapted for particular signals. For example, for a noisy signal, a dictionary composed of basis waveforms which have sharper transients or include noise might be a better choice than a dictionary composed of Gabor atoms (i.e. Gaussian-enveloped sinusoids). These basis waveforms constitute an atomic representation, or book, which reproduces the original sound. In this sense, just as Fourier analysis is the analytical counterpart to additive synthesis, DBMs can be considered an analytical counterpart for granular synthesis [9, 7].

Because atomic representations consist of potentially hundreds of thousands of atoms, it quickly becomes difficult to visualize and transform in meaningful ways. Further, there is a need to have rapid feedback with what effect a particular transformation has had on the representation. With this motivation, we have developed a prototype application, Scatter, which provides an intuitive and aesthetic interface which builds, visualizes, and transforms atomic decompositions with real-time synthesis and performance capabilities.

2. ANALYSIS

Scatter implements a GUI interface which allows a user to specify a dictionary and the terminating conditions for the matching pursuit algorithm. It also shows the progress of the matching pursuit analysis as it runs.

2.1 Matching Pursuit Algorithm

The matching pursuit (MP) algorithm decomposes a signal using an iterative gradient descent approach. At the first step, there is an original signal, a decomposed representation (the book), and a residual signal. The original signal is the signal with which we are trying to build the book. The residual signal is constructed by subtracting the book from the original signal; it is effectively the amount of the original signal which hasn't been accounted for in the representation.

At each step, MP builds a representation by choosing a basis waveform, or atom, from a given dictionary such that the waveform has the largest correlation with the residual signal. This chosen atom is then added to the book and is subtracted from the residual signal. This process is repeated until some sentinel condition is reached (i.e. some minimum residual signal or maximum number of iterations) [8].

2.2 Specifying Dictionaries

The process of specifying a dictionary for MP is flexible and somewhat of an art form. Generally speaking, any arbitrary waveform can be included in the dictionary. However, usually a dictionary element is described parametrically as a sinusoid which can be scaled, translated, enveloped and frequency modulated. Dictionaries can be made from any number of atoms of any type.

Scatter currently only allows the user to specify dictionary elements based on defining and drawing envelopes of a sinusoid. This is because dictionary elements which can be parametrically described in a general way are easier to transform according to both GUI interfaces and stochastic algorithms.

2.3 MP Implementation

Though implementation of the MP algorithm is straightforward, Scatter performs MP by using the Matching Pursuit Toolkit (MPTK) library [2]. MPTK can be used as a command line application which reads in a dictionary specified as an XML file and allows the user to specify the termination condition for the analysis. However, because of the inherent limitations of a command-line tool, it is difficult to know how well the algorithm is performing with the specified dictionary. MPTK is open source and is implemented in C++. Because of this, it was able to integrated into Scatter with a set of GUI controls for choosing the desired dictionary for a given analysis. While MPTK runs and chooses atoms at each iteration, the atom is displayed in Scatter. Though there is some impact on the running time of the algorithm, it is useful as a means to manually identify how well a given dictionary is suitable for the signal to be decomposed.

3. VISUALIZATION

Scatter visualizes books such that the energy content of atoms are accurately represented by using their Wigner-Ville distribution (WVD) [6]. The WVD is generally described as maintaining the uncertainty relationship in the perceptual discrepancy between the duration of a waveform and it's frequency perception [5, 4]. In other words, if a waveform is very short, the perceived frequency is uncertain.

3.1 Atom Colorization

Scatter also colors the atoms based on their energy contribution to the book. If an atom is overall constructively interfering with other atoms, the atom is colored blue. If an atom is overall destructively interfering with other atoms, it is colored red. An atom which is neither constructively or destructively interfering is colored black. All atoms are transparent so that it is possible to see stacks of atoms.

3.2 Visual Spatialization

MPTK has the ability to work in stereo by performing the MP over two channels. However, Scatter only uses single channel analysis to simplify the analysis and visualization implementation. However, books can be transformed spatially. This is represented visually by using a third dimension in the display. The x-axis is time, the y-axis is frequency, and the z-axis is spatial position. Because of the added dimension, Scatter's visualization can be rotated and seen from any arbitrary angles. This introduces the provocative ability to "fly" through an atomic decomposition.

4. INTERACTION

Scatter provides a variety of GUI tools that allow a user to interact with atomic representations in several ways unique to granular synthesis. There are two methods of selecting atoms in the representation: through mouse selection or through parametric selection. Once selected, atoms can be transformed either manually or by stochastic algorithms.

4.1 Mouse Selection

Scatter provides a toolbox of different atomic selection mechanisms including selection by frequency range, selection by time range, and selection by time-frequency range. Scatter also implements a lasso tool for arbitrarily-shaped time-frequency regions.

4.2 Parametric Selection

The atomic decomposition can also be selected according to any parameter range. These selections are made through range sliders. For example, all atoms of a decomposition which are within a specified duration range can be selected and transformed independent of the other atoms.

4.3 Manual Atomic Transformations

Scatter provides a toolbox which toggles different atomic transformation modes. For example, atoms can be translated in frequency and time, transposed in frequency, or stretched in duration. Also, atoms can be copied, pasted and deleted.

4.4 Stochastic Atomic Transformations

Scatter also provides a small set of stochastic transformation algorithms which modify selected atom's specified parameters according to a user-defined probability distribution function. For example, once a group of atoms are selected, the user can choose a parameter (i.e. atomic frequency or atomic spatialization) with which to modify stochastically. The amount of modification can be controlled by specifying the domain and range of a probability distribution function.

5. REAL-TIME PERFORMANCE

Scatter also has the ability to "perform" a decomposition by use of scrubbing tools. Scrubbing refers to the ability for a user to choose the location of where the real-time synthesis of the book occurs. The user can remain in one place in the decomposition for as long as desired. They can slide backwards or forwards in time at any speed they wish. The result is a performance paradigm where the atomic decomposition acts as a sonic palette.

Scatter provides a generalized scrubbing head with two parameters that can be customized by the user. The two parameters are frequency range and scrub angle. In the default mode, the scrubbing head is a vertical line which represents a precise time and over all frequencies. Any atoms over the entire frequency range of the book which intersect with the scrub head at that time will be synthesized at their appropriate phase and duration offsets within their individual envelopes.

If the frequency range is changed, only atoms within the specified frequency range will be synthesized. If the scrub angle is changed such that, for example, the slope of the scrub head is positive, then atoms in a book which have higher frequencies will be synthesized before atoms of lower frequencies.

6. CONCLUSIONS

Scatter is a tool which allows a user to analyze, visualize, transform, and perform atomic decompositions. It is still in it's prototype phase and has many issues which still need to be addressed such as for example, the adoption of a standard MP book file format, dealing with arbitrary basis waveforms, and allowing for more customization in the details of the MP algorithm.

7. ACKNOWLEDGMENTS

Thanks to MAT Ph.D. students Lance Putnam and Graham Wakefield for consulting on various implementation details of Scatter. This work was supported in part by the National Science Foundation under Grant CCF-0729229.

8. REFERENCES

[1] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, Aug. 1998.

[2] S. Krstulovic and R. Gribonval. MPTK: Matching pursuit made tractable. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, volume 3, pages 496–499, Toulouse, France, Apr. 2006.

[3] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, Dec. 1993.

[4] P. Masri, A. Bateman, and N. Canagarajah. The importance of the time-frequency representation for sound/music analysis-resynthesis. *Organised Sound*, 2(3):207–214, 1997.

[5] P. Masri, A. Bateman, and N. Canagarajah. A review of time-frequency representations with application to sound/music analysis-synthesis. *Organised Sound*, 2(3):193–205, 1997.

[6] D. Preis and V. C. Georgopoulos. Wigner distribution representation and analysis of audio signals: An illustrated tutorial review. *J. Audio Eng. Soc.*, 47(12):1043–1053, Dec. 1999.

[7] C. Roads. *Microsound*. MIT Press, Cambridge, MA, 2001.

[8] B. L. Sturm, C. Roads, A. McLeran, and J. J. Shynk. Analysis, visualization, and transformation of audio signals using overcomplete methods. In *Proc. Int. Computer Music Conf.*, Belfast, Ireland, Aug. 2008 (submitted).

[9] I. Xenakis. *Formalized Music*. Indiana University Press, Bloomington, Indiana, 1994.