Techniques for Multi-Channel Real-Time Spatial Distribution Using Frequency-Domain Processing.

Ryan H. Torchia & Cort Lippe

Hiller Computer Music Studios, Department of Music, University at Buffalo
email: torchia@buffalo.edu, lippe@buffalo.edu

Abstract

The authors have developed several methods for spatially distributing spectral material in real-time using frequency-domain processing. Applying spectral spatialization techniques to more than two channels introduces a few obstacles, particularly with controllers, visualization and the manipulation of large amounts of control data. Various interfaces are presented which address these issues. We also discuss 3D “cube” controllers and visualizations, which go a long way in aiding usability. A range of implementations were realized, each with its own interface, automation, and output characteristics. We also explore a number of novel techniques. For example, a sound’s spectral components can be mapped in space based on its own components’ energy, or the energy of another signal’s components (a kind of spatial cross-synthesis). Finally, we address aesthetic concerns, such as perceptual and sonic coherency, which arise when sounds have been spectrally dissected and scattered across a multi-channel spatial field in 64, 128 or more spectral bands.

1 Introduction

Real-time spectral processing is now so commonplace that patches employing FFT analysis/resynthesis techniques to do parametric-like EQ and cross-synthesis are included as stock examples in such software as the popular Max/MSP package, SuperCollider, etc., (Settel and Lippe 1997). Those methods provide a starting point for the development of techniques for spatially distributing spectral material. Since our implementations are in Max/MSP, all the examples in this paper are described in that environment (Zicarelli 1997).

2 Stereo distribution of spectral material

A simple, user-controlled method of distributing spectral material in a stereo field is the easiest way to introduce the fundamental concepts behind spectral spatialization techniques (Settel and Lippe 1994). Since the only parameters we need to control are FFT bin number and left-right location, a simple two-dimensional table (here shown as a Max/MSP *multiSlider*) is used as the primary controller. For our purposes, the *multiSlider* has 64 individual sliders representing FFT bins from low to high; each slider has a resolution of 128 (0-127 for left-right). Figure 1 illustrates this basic controller. (It should be mentioned that we use FFT window sizes much larger than 64 samples for sound examples associated with this paper.)

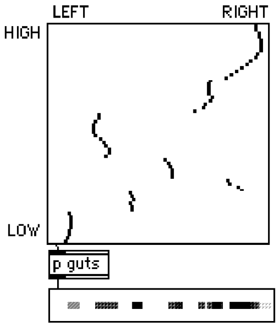

Figure 1: A *multiSlider* controller for two-channel distribution of spectral material.

The output of the *multiSlider* controller is fed into a subpatch (figure 2), which converts each slider value into left and right volume coefficients ranging from 0-1. We made use of a subpatch borrowed from the MSP MIDI panning tutorial for much of our work, which maps the values for speaker-to-speaker panning (Dobrian 2000). (This is an arbitrary choice of mapping algorithm; others produce effective results as well.) The speaker-to-speaker calculations are performed independently for each of the 64 sliders. The two sets of values (one for left channel and its inverse for the right channel) are stored using their respective FFT bin number as indexes in separate signal buffers using the *peek~* object.

Figure 3 shows a simple *pfft~* subpatch, in which the spectral information from a monophonic input signal is multiplied separately with each spatial location signal buffer previously generated; each channel is then resynthesized independently. Thus the spectral material from the monophonic input signal is distributed out the left channel (*fftout~ 1*) and/or right channel (*fftout~ 2*) as specified in the *multiSlider*.

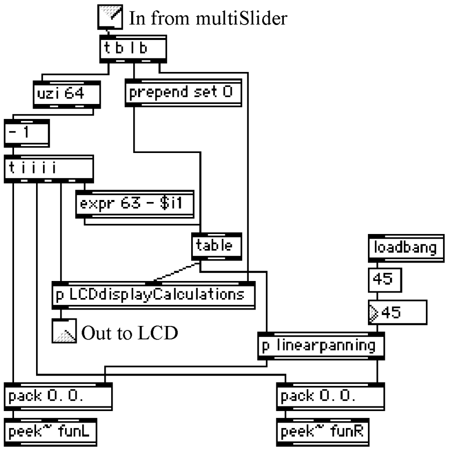

Figure 2: A subpatch showing conversion from the *multiSlider* controller to twin spatial information tables. The speaker-to-speaker panning subpatch is borrowed directly from the MSP panning tutorial.

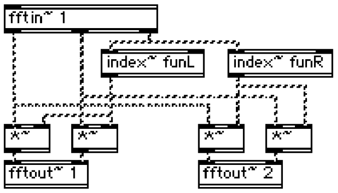

Figure 3: The *pfft~* subpatch showing lookup in the two spatial information tables for stereo spectral spatialization.

3 Four-channel distribution of spectral material.

Expanding the two-channel spectral spatialization technique to four channels or more is a fairly simple process. A four-channel system requires the addition of two additional transfer tables (*peek~* objects) and two additional *fftout~* objects. However, this introduces fundamental problems regarding input and output interfaces and representation. The ability to simultaneously specify the control of three distinct parameters (i.e. a spectral point in 2D space) is of particular concern. As a practical solution, a second *multiSlider* was added to our interface to control front-back spatial distribution. While this solution allowed full control over each spectral band in a two-dimensional space, it requires that each spatial dimension be set separately (see figure 4). Obviously, three-dimensional controllers—outside the scope of this paper—have great potential in this area.

Another control-related issue introduced by the additional spatial dimension has to do with visual feedback, specifically, how to indicate the spatial location of 64 (or more) independent bands of spectral information on a two-dimensional screen. One possible solution involved the introduction of color to represent the FFT energy bins (as shown in the bottom half of figure 4). By scaling FFT bin numbers to RGB values, the Max/MSP *lcd* object can be used to show the location of each bin in two-dimensional space. In addition, the authors created an intuitive visual feedback tool using Jitter, a set of objects for Max/MSP, which includes 3D graphics capabilities (Clayton 2002). This appears to be a very satisfactory solution for output representation.

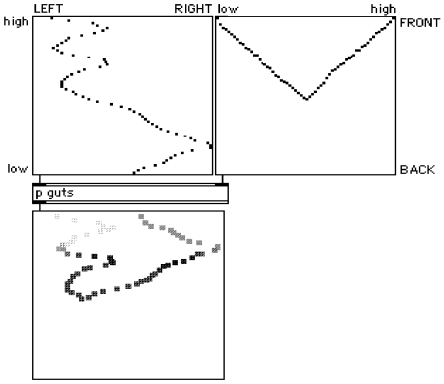

Figure 4: A spectral spatializer controller using separate *multiSlider* objects for left-right and front-back localization. The *lcd* display indicates the location of FFT energy bins in two-dimensional space. Color is used to indicate the relative frequency of each bin.

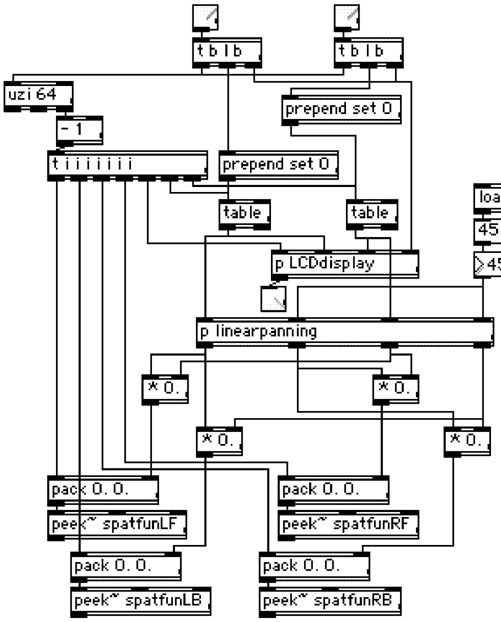

Figure 5: The four-channel version of the subpatch that converts two sets of *multiSlider* values into amplitude coefficients for four spatial information tables. The linear panning subpatch contains two copies of the stereo panning algorithm, the output of which is cross-multiplied to generate the four amplitude coefficients for localizing a monophonic sound in quadraphonic space.

4 Advanced applications

The controllers, interfaces, and techniques discussed thus far illustrate the basic idea, and are useful in a wide variety of applications. Of course, any element of the spatialization process can be tailored for specific results. The authors experimented with a number of control interfaces and spatialization techniques, some of which will now be discussed.

4.1 Circular spatial distribution

With only minor changes to the spatialization algorithms and/or controllers, several new techniques for spectral spatialization can be created. One example is an algorithm that results in the distribution of sound in a circular or geometric pattern around a center point in the listening space. Every bin is given a different spatial location. (Using 64 bins, the circle or pattern is described by 64 points.) By adding additional mathematical operators to the algorithm, the shape of this pattern and the energy distribution of the signal (the "spread" or radius) can be altered in real time. To compliment this algorithm, the controller was redesigned to benefit the way the sound wraps around the listening space, and automation was added which allows the spectrally diffused sound to be "spun" in various ways around the audience.

4.2 Spatialization based on signal analysis

Another area of great interest to the authors involved using information derived from spectral analysis of audio signals to control spectral spatialization. Several techniques were developed which allow selected spectral characteristics (e.g. the amount of energy or phase position per bin) of a sound to create a spectral spatialization pattern. The sound used to determine the spatialization pattern could be the same signal actually being spatialized (a kind of "self-spatialization"), or a different signal altogether ("spatial cross-synthesis").

In both the self-spatialization and spatial cross-synthesis models, we have experimented with a variety of mappings, which yield interesting results. For example, in one mapping, high-energy bins are grouped towards the center of the spatial field, while low energy bins are distributed farther from the center; the inverse of this also provides significant results.

Certainly, the choice of sounds that lend themselves to being effectively spatialized, and the choice of sounds whose spectral characteristics offer interest as spatial controllers is of the utmost importance. Ultimately, approaches based on signal analysis still require an arbitrary decision on how the selected property of the sound is to be transmogrified to spatial coordinates.

4.3 Stereo spatial cross-synthesis

A method was devised for mapping the energy dispersion pattern of a stereo signal (the modulator), and then applying that pattern to an incoming monophonic signal (the carrier). Two channels are brought into the frequency domain, and for each FFT bin, the amount of energy in each channel is compared and weighted on a scale from zero to one for both left and right. These relative weightings are then used as amplitude coefficients for a monophonic signal in the frequency domain, which is then resynthesized as a stereo signal (see figure 6). This technique is similar in many ways to the design and sound of conventional cross-synthesis, especially if the stereo signal used includes definite, perceptible events panned hard to one channel. However, unlike cross-synthesis, all frequencies present in the original monophonic signal will still be present at the output, regardless of whatever is found in the analyzed stereo signal. Since the stereo signal is only analyzed for spatial energy distribution, silence in the modulator signal is essentially the same as any other equally balanced signal, and results in the monophonic signal remaining monophonic. This technique is rife with opportunities for additional spectral processing, such as filtering out frequencies that are not significantly panned hard right or left, or exaggerating the stereo spectrum.

In addition, another use of a stereo signal as a spectral controller is much like the monophonic spatial cross-synthesis technique described above. In this case the left-right energy can be used in more arbitrary fashion than that described in the preceding paragraph.

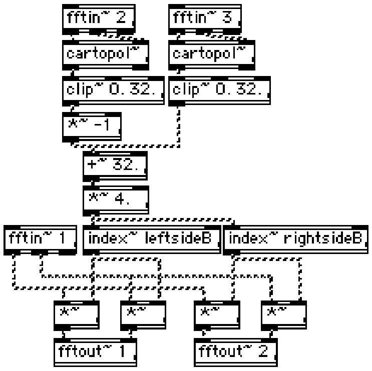

Figure 6: The *pfft~* subpatch used to multiply the spatial pattern of a stereo signal onto a monophonic signal.

5 Practical and aesthetic considerations

5.1 Automated controllers

All of the techniques discussed thus far can operate in real time using Max/MSP. The complexity of controlling the processes and the limitless possibilities available makes some implementation of presets and automation attractive for performance; however, a thorough discussion of automation techniques is outside the scope of this paper. Since the *multiSlider* object is fairly common in spectral synthesis applications, automation created to manipulate it would likely be useful in a variety of patches.

5.2 Aesthetic issues

As mentioned above, and as with other forms of spectral processing, choice of sonic materials is key to employing spectral spatialization effectively. (For instance, materials that are rich in higher frequencies seem to work well in many cases.) Spectral spatialization can also be employed in tandem with other forms of signal or spectral processing, such as filtering/noise reduction; the paragraph below describes a particularly interesting application of this.

Another issue outside the scope of this paper is one of reference: a sound that has been spectrally dissected and scattered across a spatial field may lose its perceptual coherency as a single sonic event, and may instead appear to the listener as several unrelated sounds. Possible solutions for this problem include using sonic materials that maintain cohesiveness throughout the spectrum (such as a speaking voice, clear sonic lines, or some pitched percussion), or the gradual introduction of spatialization, such that the audience is made aware of the process as it occurs might be used.

The two- and four-channel techniques discussed above may also be used as a spectral signal router. For instance, simply by sending the signals for each separate channel to additional processing modules, either in the spectral domain or after time domain conversion, different spectral regions of a sound can be processed differently. In this context, the number of channels may be greatly increased. The controller is effectively converted from a spectral spatializer to spectral auxiliary sends.

6 Conclusion

Combining spectral analysis and resynthesis with spatialization offers several new possibilities for multi-channel composition and performance. The general concept lends itself to a variety of implementations limited only by the imagination. Spectral spatialization can be employed in pre-recorded compositions or real-time performances, and can be used independently or in conjunction with a variety of other effects.

7 Future Directions

We are continuing to develop new spatialization algorithms, and create a library of automation for existing algorithms in order to be able to combine flavors of spectral processing quickly and easily. In addition, we plan to combine some of this research with earlier work in the area using low frequency time-domain signals in the frequency domain to control spatialization (Settel and Lippe 1998). We are also experimenting with spectral delay techniques that add an interesting dimension to spectral spatialization. Defining both a spectral location and delay time (with or without feedback) for a specific bin offers unique possibilities to produce certain spatial techniques involving distance, etc. In addition, interpolation between spectral frames can slow down the rate of change of spectral data, producing bin-specific reverberation effects.

8 Acknowledgments

The authors would like to thank Miller Puckette, David Zicarelli, and Convolution Brother #2 for their input and support.

References

Clayton, J. 2002. Jitter Software. San Francisco, Cycling ‘74.

Dobrian, C. 2000. “Tutorial 22: MIDI panning.” *MSP: Tutorials and Topics.* San Francisco: Cycling ‘74.

Settel, Z., and C. Lippe. 1994. “Real-time Timbral Transformation: FFT-based Resynthesis.” *Contemporary Music Review* 10: 171-179.

———. 1997. “Forbidden Planet” and “Cross Dog AB.” *Max/MSP Examples.* San Francisco: Cycling ‘74.
San Francisco: Cycling ‘74.

———. 1998. “Real-time Frequency Domain Signal Processing on the Desktop.” *Proceedings of the 1998 International Computer Music Conference.* San Francisco: International Computer Music Association, pp. 142-149.

Zicarelli, D. 1997. Max/MSP Software. San Francisco: Cycling ‘74.