ISEA2015
Proceedings of the 21st International Symposium on Electronic Art
ISSN: 2451-8611
ISBN: 978-1-910172-00-1

Extracting Visual Information to Generate Sonic Art Installation and Performance

Eric Heep Ajay Kapur

California Institute of the Arts, Music Technology: Interaction, Intelligence, and Design
California, United States
ericheep@alum.calarts.edu

Abstract

A procedure for generating sound using visual information is outlined that allows for a data artist to interpret a visual work of art using the parameters of an Inverse Discrete Fourier Transform. This paper discusses the historical progression of musicians responding to visual artists, as well as the relevance of parametric articulation and how it relates to the science of audio analysis. A process is outlined that discusses how such techniques can be used to generate sonic art installation and performance.

Keywords

Music Information Retrieval, Sonification, Audification, Inverse Discrete Fourier Transformation, Procedural Composition, Parametric Articulation

1. Introduction

There is a historical tradition of composers drawing inspiration from visual art. Modest Mussorgsky's "Pictures at an Exhibition" was famously based on the work of artist Viktor Hartmann. Mussorgsky planned to "draw in pictures" the watercolors and drawings of his recently deceased friend. [1] Morton Feldman emulated the work of the abstract expressionists, which inspired the composer to attempt a music that was "more direct, more immediate, more physical than anything that had existed heretofore." [2] These composers and others like them were working with the inspiration drawn from art, and in turn, exploring how to directly transform visual works into sound.

Our motivation was drawn from this tradition, as we utilized a more direct way to generate music and sound based on visual artworks. While the human element of translating visual art into sound may be partially obscured by procedural techniques, the technique outlined still allows for ample subjective control of the resulting sound.

Other applications do exist that can sonify digital images in real time. SonarX, while originally meant to aid blind users, has found applications for art and performance. [3] SonarX has mappings for pitch scale, timber, and other musical parameters. This paper does not wish to delegitimize this use, but instead offers a different way of sonification that uses only the IFFT itself.

This paper will first discuss the use of the Inverse Discrete Fourier Transform and its legitimacy as an interpretive tool. It will then outline the process of generating sound from images and discuss any technical or subjective considerations that arise. Finally, it will discuss the aesthetics of the discussed method, as well as the application of such a process for installation and performance.

2. Inverse Fast-Fourier Transform and Composition

A direct translation from image to sound is available using an Inverse Fast-Fourier Transform (IFFT), as sound can be generated directly from the visual data of various works of art. To explain the motivation behind using an IFFT on images, the process of using a Fast-Fourier Transform (FFT) on audio must first be touched upon.

It is common in the field of Music Information Retrieval to utilize a Fast-Fourier Transform to gain meaningful information from a piece of audio. Likewise, it is also common to display this information as an image that represents the frequency content of the audio, see Figure 1.

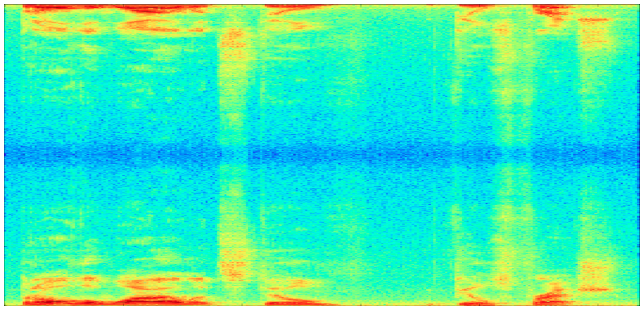

Figure 1. A spectrogram of a vocal excitation containing the words "All the while it still..".

The above image is a frequency domain representation of audio. A Fast-Fourier Transform is utilized to analyze several frames of the audio, which are then sequentially ordered in the image above.

Even though by using a reversal of this process it is possible to reconstruct the audio from the information that was extracted from the original audio, it is also possible to utilize this same reconstruction process starting with visual data that was not originally audio. This is done by using an Inverse Fast-Fourier Transform to turn images into sound instead of using the Fast-Fourier Transform to turn sound into images.

Our motivation behind this method of sound generation is based on the compelling visual similarities between spectrograms and the work of modernist painters. A natural curiosity arises to imagine how a Mark Rothko or a Jackson Pollack painting might sound. While a spectrogram is not the only way to visually represent sonic data, it is germane when comparing modernist paintings to a visual representation of sound.

This method of sound generation is a natural extension of the ones used by composers who have relied strictly on process and acoustics to guide their compositions. James Tenney would compose his music according to a process "for the sake of perceptual insight," as many of his compositions were guided by acoustics. [4] Directly translating works of art using an IFFT follows a similar procedural method and allows for a glimpse at the perceptual insights that Tenney sought.

In the music of Tenney, parametric focus and parametric articulation surpass pitch, as he stresses "the greater importance that has been given in 20th-century music to *all* the parameters of musical sound." [5] In this tradition, parametric articulation stands alongside other aspects of music such as pitch, texture, or rhythm. Following in Tenney's footsteps, conventions can be established to stay within the confines of a process when utilizing an IFFT, which are determined by the program in which the data artist utilizes an IFFT. Examples of such options are the amount of frame overlapping, the IFFT-size, or the type of windowing. These will be discussed more in depth further in the paper, but for now serve to elucidate how the technical parameters of the IFFT become the compositional and interpretive methods of the construction process.

After the IFFT procedure is complete, no post-processing of the resulting audio is done. The IFFT is sufficient for the data artist to who wishes to create a direct translation of a visual work. We posit that there is enough interpretive control given the parameters of the IDFT that there is no need for further interference. Since the procedure involved is the IDFT itself, any further modification of the of the audio would take focus off this procedure. The resulting audio could be characterized as raw and direct. Because our source material stems from modernism, it is natural to also draw from their ideals. Frank Stella famously described his work with the quote, "My painting is based on the fact that only what can be seen there is there.. What you see is what you see." [6] In our case, the audio is based on the fact that only what can be seen is heard, what you see is what you hear.

3. System Design

A digital scan or photograph of a painting exists in computer memory as a collection of pixels. Because of the manner in which audio is extracted from visual data, the size of the image must be taken into account. The resolution of the image corresponds with the amount of audio generated; a larger image will generate more audio content than a smaller image.

To put it in terms of an IFFT, the IFFT-size is linked with the amount of samples produced. An IFFT-size of 1024 will produce exactly 1024 samples, while an an IFFT-size of 256 will produce exactly 256 samples.

Considering that common IFFT sizes are generally powers of 2 (256, 512, 1024), it will most likely be necessary to resize the image. The data artist has a choice as to what resolution they wish to resize their image to, knowing that the pixel height they choose will determine the length of their composition.

Color Space Considerations

Because color images have three channels of data (red, green, and blue), further subjective choices are presented when deciding on how to interpret the red, blue, and green value of each pixel. Outlined are multiple methods for handling the three channels.

The first method consists of reducing the three color channels down to a single grayscale channel. The first and simplest choice is the luma equation, which is an average of the three channels with weighting coefficients, shown in Figure 2. [7]

$$Y = 0.21\mathrm{R} + 0.72\mathrm{G} + 0.07\mathrm{B}$$

Figure 2. Luma grayscale equation.

This is more perceptually accurate than simply averaging all three channels, but it is possible to be more perceptually accurate by converting to a CIE XYZ color space and using the "luminance channel as a grayscale representation of the original color image." [8] The luminance channel being Y in Figure 3.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} * \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Figure 3 CIE Y grayscale equation. [9]

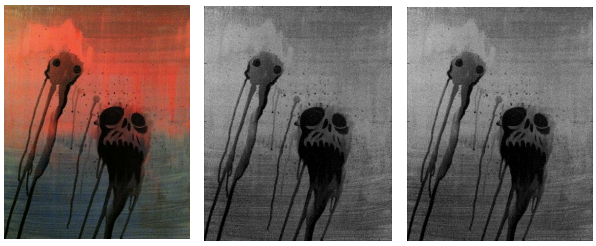

Figure 4. The luma grayscale conversion (middle) and CIE Y grayscale conversion (right) of Sean Ryan's "Soggy Spirits" (left), printed with permission.

Because the three channels have been reduced to one, we are also reducing the resulting number of audio channels to one. While a grayscale conversion loses the

original color data, necessary information about the image is retained and an accurate aural representation can be produced that is tied to human perception.

There is also the option to not convert to grayscale at all, and instead give each color channel its own audio channel. This route could be considered the most perceptually inaccurate of the three. This approach also grants the freedom to convert to any color space the data artist sees fit, and allows for various spatialization techniques because there are more audio channels resulting from the IFFT process.

Mapping Color Components

Since our RGB values range from 0 to 255 and the magnitudes of the IDFT algorithm range from 0.0 to 1.0, we have the option of mapping our color components linearly from the interval [0,255] to the interval [0.0, 1.0]. This preserves our grayscale conversion and is the most straightforward approach.

We have also explored using standardization instead of linear mapping, and found it useful in limiting the influence of outliers in our spectrum. The equation is shown in Figure 5.

$$x_n = \frac{x_n - \bar{x}}{\sigma} \qquad \sigma = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_n - \bar{x})^2}$$

Figure 5. The standardization, which is the removal of the mean divided by the standard deviation.

If we standardize, we ensure that the spectrum is not skewed considerably by a datum that is significantly larger than the others. The major drawback of standardizing our data is that we begin to stray from our goal of creating a direct sonic portrayal of the visual work.

Inverse Fast-Fourier Transform

An IFFT usually reads from complex numbers, which consists of magnitude and phase. We will be mapping our color components to the magnitude input of the spectrum, and reading our image left-to-right with a left-bottom origin.

An option can be taken to create artificial phase values to feed into the IFFT alongside the visual data. This step is not entirely necessary, but remains a subjective decision for the data artist.

The hop size is the amount of overlap present between sequential IFFT frames. Hop sizes are generally a fraction of the IFFT-size, and are another consideration for the data artist in determining the length of a composition. For example, a hop size that is 50% of the IFFT-size will result in a composition that is half the length of a composition with a hop size of 100% (or no overlap).

Windowing is then performed to envelope the individual audio frames generated by the IFFT. A variety of envelopes were constructed based on the various window functions used in an FFT. Subjectively, we found the Parzen window to sound the best when applied to the audio generated from the IFFT. The Hann, Blackman-Harris, and Exponential window functions also performed well. The equation for the Parzen window is provided in Figure 6.

$$w(n) = \begin{cases} 1 - 6\left(\frac{|n|}{N/2}\right)^2 + 6\left(\frac{|n|}{N/2}\right)^3 & n \le |n| \le (N-1)/4 \\ 2\left(1 - \frac{|n|}{N/2}\right)^3 & (N-1)/4 \le |n| \le (N-1)/2 \end{cases}$$

Figure 6. The equation for a Parzen window, which is to be used on the audio frames extracted from the IFFT. [10]

4. Aesthetics and Practice

The resulting audio is extremely raw, but a significant amount of information is heard. Each "row" of the painting corresponds to a center frequency bin of the IFFT, which provides sound throughout the entire audible spectrum (granted program's sampling rate is set to 44100 or above).

Because the source material is treated with a limited amount of interpretation, the data artist has created a system that allows the painting itself to be heard instead of only hearing the artist's interpretation of the painting. Thus the meaning found in the sonification is shared between the artist and the painting. This point is salient, considering that sonification is "concerned with the creation of representations of data that facilitate inference and meaning making." [11]

Both installation and performance are available to the data artist. Off-line computation is done for installation work, with the audio being presented alongside a visual projection of the artwork used for analysis. Multiple works of the same visual artist are typically presented by the musician, ensuring a thematic continuity. The paintings of Mark Rothko have been a favorite for their striking similarity to spectrograms. The technical parameters of the audio are altered per piece, and tailored to each image. These parameters can be fine-tuned to highlight the aural differences that arise from the visual differences between these visual works, and more importantly, can also be seen as an artist's interpretation of the work. The Python programming language is used for off-line sonification, with the data manipulation possible using the Numpy and Scipy modules.

Real-time performance by an artist is also available, the variation of the technical parameters becoming paramount to the performance. This allows the artist to respond to the sonification in real-time. Performance is made possible by the music programming language, ChucK. [12] A real-time IFFT is utilized, and dynamic control of its parameters are available to the performer. An excerpt of this code is shown in Figure 7.

```
fun complex[] calcWindow(float frame[], float mag, float
phase)
{
   complex X[frame.size()];

   // loop for creating a fram
   for (int i; i < bins; i++) {
        // phase incrementing
        (ph_inc[i] + ph[i]) % tau => ph[i];
        polar temp;

        // color components being assigned
        Math.fabs(frame[i]) * mag => temp.mag;

        // artificial phase being assigned
        ph[i] * phase => temp.phase;

        // result to be ChucKed to the
        // IFFT.transform
        temp $ complex => X[i];
   }

   return X;
}

fun float[] playWindow(complex X[], dur window) {
   0 => X[0];
   // inverse fft that reads the window
   ifft.transform(X);

   // dividing to envelope length
   window/2.0 => env;

   // sets attack and release of
   // the window function envelope
   win.setParzen();
   win.attack(env);
   win.release(env);

   win.keyOn();
   env => now;
   win.keyOff();
   env => now;
}
```

Figure 7. ChucK code including two functions. calcWindow() implements artificial phase by phase-incrementing a series a sine waves corresponding to the frequency bin it belongs to. playWindow() sets the windowing function type (Parzen in this case) and plays the inputted frame of visual data.

An understanding of the various IFFT parameters is necessary for performance, with parametric articulation allowing for expressive control over the general rhythm and timbre of the piece. A reaction to the paintings by the data artist is heard in real time, allowing both a direct transformation of the visual data into audio, as well as a raw interpretation of the painting by the performer.

This interpretation is heard in how the performer allows the IFFT to translate the data. A Triangular window could be used instead of a Parzen window in if the performer requires a rougher sound, or the sound can be made sparse by decreasing the rate at which the data is read.

The overall process allows the data artist individual expression while still abiding to a principled process. The manner of translation and parametric articulation continues upon a compositional tradition, and allows for a direct yet interpretive translation of visual art to sonic art.

References

1. Michael Ross, *Mussorgsky: Pictures at an Exhibitiion* (University of Cambridge, 1992), 16.
2. Alex Ross, *The Rest is Noise* (Picador, 2007), 529.
3. M. Mengucci, F. Medeiros, and M. Amaral, “Image Sonification Application to Art and Performance”, accessed May 20, 2015 http://users.fba.up.pt/~mc/ICLI/mengucci.pdf
4. James Tenney, “Interview”, *The Postal Pieces* CD (New World Records, 2004), liner notes.
5. James Tenney, *Meta-Hodos and Meta Meta-Hodos* (Inter-American Institute for Musical Research, 1988), 16-17.
6. Alex Potts, *The Sculptural Imagination: Figurative, Modernist, Minimalist* (Library of Congress, Cataloguing in Publication, 2000), 272.
7. Charles Poynton, *Digital Video and HDTV: Algorithms and Interfaces* (Morgan Kaufman Publishers, 2003), 207.
8. M. Cadik, “Perceptual Evaluation of Color-to-Grayscale Image Conversions”, *Computer Graphics Forum* 20-7 (2008): 1745-1754.
9. Hugh Fairman, Michael Brill, Henry Hemmendinger, “How the CIE 1931 color-matching functions were derived from Wright-Guild data”, *Color Research and Application,* 21(1988): 11.
10. Fredric Harris, “On the Use of Windows for Harmonic Analysis With the Discrete Fourier Transform,” *Proceedings of the IEEE* (January, 1978) 66. http://web.mit.edu/xiphmont/Public/windows.pdf
11. Barass and Vickers, “Sonification Design and Aesthetics”: *The Sonification Handbook*, (Logos Publishing House, 2011) 154.
12. “ChucK : Strongly-time, Concurrent, and On-the-fly Music Programming Language”, http://chuck.cs.princeton.edu/, accessed May 24, 2015