Xenakis's Matrix to Image Matrix:
A Sonification Method Inspired by *Achorripsis*

Serkan Sevilgen, Department of Music and Performing Arts, Yildiz Technical University, Turkey,
ssevilgen@gmail.com – https://serkansevilgen.com

Proceedings of the *Xenakis 22: Centenary International Symposium*Athens & Nafplio (Greece), 24-29 May 2022 - https://xenakis2022.uoa.gr/

Abstract

This paper presents research on data sonification that primarily focuses on digital image data to compose algorithmic music. The main focus is escaping from the bottom-up approach that is ingrained in generative music and building the ground for multiscale formal strategies on sonification. The proposed composition method is inspired by the theories and techniques developed by the composer Iannis Xenakis (1922 - 2001). He creates a matrix for his piece Achorripsis (1957) with columns that represent time and rows are groups of timbres. He employs the Poisson formula to fill the cells with values to manifest his macrocompositional design. Even though he uses a probabilistic distribution method for his algorithmic way to generate music he managed to avoid the pitfalls of bottom-up strategy and has control over the form of music as a composer. The matrix has a strong resemblance to a digital image which is a collection of pixel values in a rectangular shape. Even further we parse a digital image file and convert it to an actual matrix in the programming environment. An image has its own formal structures that we can observe easily. Each cell which is a pixel or a group of pixels is already filled with data that can be subjected to sonification. In this research to preserve the high-level relations that occurred in images, a machine learning algorithm is used for segmentation. Selecting few segments generates large chunks of data which helps us to define high-level, form-defining elements. The procedure is repeated with a higher number of segments to let us make more granular-level decisions. Our method is backed by a bespoke computer program written by the author to streamline technical processes and aims to be flexible to support diverse artistic ideas. The software is built on Python programming language (the leading platform for data analysis and machine learning) and processes input data to generate audio with Csound audio programming platform. Using a digital image in this method has the following advantages over other types of data sonification: The structure of image data is well-defined and helps us to automate data-specific laborious tasks like cleaning, fixing, and restructuring that makes the research repeatable with different sources conveniently; an average size image file provides sufficient data points; it is easy to access a wide variety of image files; the current research on digital image analysis and processing (including machine learning) has the potential to create new opportunities for our work. The musical work will exemplify the main workflow and mapping strategies to handle multiscale strategies will be discussed.

1. Introduction

Data sonification is a technique of translating data into non-speech sounds. (Kramer, et al. 1999) It can be regarded as a scientific procedure that prioritizes accuracy and tries to achieve objectivity or artistic practice which seeks convincing narrative and relies on subjective matters. Considering sonification as a tool in algorithmic composition or generative music, (Spiegel 2018) helps us to put that effort into musical perspective. Many sections of algorithmic music that utilize the mapping data value onto musical parameter suffer from the problem we present here in the sonification domain: The process of composing bases on bottom-up approach (Roads 2015) which musical form is expected to reveal itself over time depending on the fluctuations of data. Composers abstain from constituting macro-level organizations. This approach works very well in large time scale works with days of duration and sound art which are not considered to be consumed by an audience from start to finish. But if the composition is assumed to listen to in one sitting then the audience should work with their memory to construct and relate sections of music temporally. Our proposal is a multi-scale composition model that employs image data sonification with inspirations from theories and works by Iannis Xenakis. We present the preliminary results of the experiment to demonstrate how the model can be applied to compose electroacoustic music.

Supper (2001) presents that composers deliberately build the formal structures by constructing algorithms for their musical expression needs either to realize established tonal or serial music like Lejaren Hillar and Koenig did or a novel form as Clarenz Barzow in his piano piece *Cogluotobusisletmesi.* In both cases, the algorithms are only a tool to help composers. On the other hand, he examines the use of extra-music algorithms like L-systems derived from chaos theory by the composer Hanspeter Kyburz. In this case, the form is the direct outcome of the algorithm without leaving too much to the composer. Roads (2015) discusses the design strategies regarding material-form relations. "The top-down approach starts by predefining a macroform, a template whose details are filled in at later stages of composition." Traditional forms like sonata, rondo, etc. can be considered as an example of the top-down approach. They are a set of rules that composer creates suitable sound materials. On the other hand, the bottom-up approach "constructs form as the final result of a process of internal development produced by interactions on low levels of structure" Sequence layering, indeterminacy, and generative algorithms are examples of the bottom-up approach. Roads proposes the multiscale strategy that the composer can employ both approaches moving in different (micro and macro) timescales according to the needs of the composition process.

Nick Collins (2009) states that form was not one of the main concerns in algorithmic music. Yet he lists the following generative techniques that might result in novel formal structures:

1) Self-similar structures; recursive definitions, fractal music;
2) Stochastic music, information theoretic constructions;
3) Emergent form (as a by-product of complex lower level activity);
4) Artificial musical grammars.

We can argue that "self-similar structures" and "emergent form" can be considered as a part of the bottom-up approach whereas "artificial musical grammars" as top-down. Iannis Xenakis shows that in "stochastic music" a composer can employ both approaches that link us to multiscale form building strategies as we investigate in this paper.

2. Image data

In order to build a composition model that is based on sonification, we need relatively large data structures so that we can map as many sound parameters, and create mid to large-scale works. A digital image file is made of pixels that are defined as RGB (Red, Green, Blue) values. A 24-megapixel camera captures 24 million pixels with each one holding three values for Red, Green, and Blue which yields 72 million data points. For reference, a simple audification (direct converting data to audio) of that data would produce 27 minutes of audio at a 44.1 kHz sampling rate. The resources are abundant and it is easy to access or create digital images that can help to replicate the experiments with different data files. Since image data is already a visualized form of data, browsing for it and further exploration is easy and intuitive. The structure of the digital images is well established. JPEG specification which is a widely used compressed image format was first published in 1992 (Pennebaker & Mitchell 1992). Software packages and libraries are developed to parse, convert and manipulate different image formats. This capacitates the automation of laborious data preprocessing for successive experiments. However, the sonification of an image is tricky due to its formal structure and perception that differs wildly from music.

2.1. Achorripsis

Achorripsis is a piece composed in 1957 and can be seen as the realization of the Phases of Composition idea. The piece was composed for 21 instruments which were grouped in seven timbre categories: Flute (piccolo, clarinet, bass clarinet), oboe (oboe, bassoon, contrabassoon), brass (two trumpets, trombone), percussion (xylophone, woodblock, bass drum), pizzicato, string glissandi, string arco (3 violins, 3 violoncellos, 3 contrabasses). He constructed a matrix with 7 columns that each one is dedicated to one of the timbre groups. On the horizontal axis, he used 28 rows as equal time units. The music is planned to endure 7 minutes then each time unit is 15 seconds. Each cell of the matrix was filled with event definitions, like no event, single event, double event, triple event, and quadruple event. Events determine the density of a timbre group at a particular time. The total number of event types was determined by using the *Poisson* formula. The *Poisson* formula is a mathematical equation that calculates the probability of an event based on the Poisson distribution. Xenakis used different probability distributions in his compositions like *Poisson, Cauchy, Gaussian*, and *Uniform* distributions. "*Poisson* distribution is used to model processes where the distribution

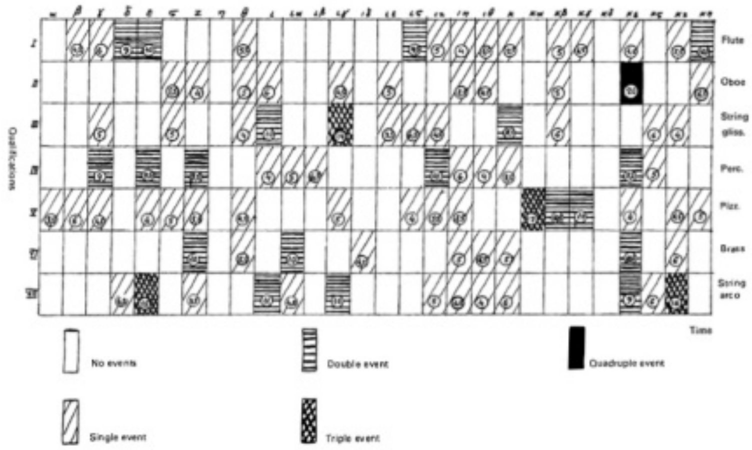

Figure 1: The matrix that Xenakis designed for Achorripsis

of the number of incidents occurring in any interval depends only on the length of that interval" (Attenborough, 2003). The *lambda*, mean value of the distribution was arbitrarily chosen as 0.6 to produce at least one quadruple event, 4 triple events, 19 double events, 65 single events, and 107 zero events. in the 196 cell matrix. The events were spread into cells again using the Poisson formula. We can follow phases in the compositional realization of Achorripsis. Xenakis started with initial ideas (timbres, density of clouds) then created a macro-compositional model with the matrix. From top to down he meticulously calculated every event until he reached the symbolic output of the music which is notation. The formal structure is not emerged due to the musical mapping of stochastic calculation on to basic vector of a note as pitch, duration, and amplitude. "Poisson distribution is used to model processes where the distribution of the number of incidents occurring in any interval depends only on the length of that interval" (Attenborough 2003).

3. Xenakis' matrix to image matrix

The pixels are arranged as rows and columns in a typical image file. When we parse an image file in a programming environment we create a matrix structure with rows and columns. Reversely we can generate a matrix and then visualize it. If we wish to create an image with 7 rows height and 28 columns width and assign event values like zero, one, two, three, and four to each cell and color-code them we end up in a two-dimensional space that resembles Xenakis's matrix for Achorripsis.

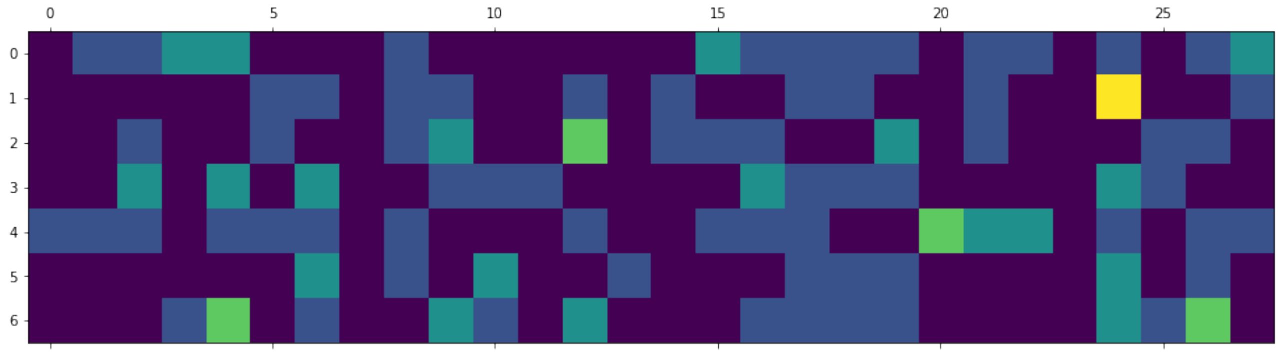

Figure 2: Achorripsis Matrix recreated with color-coding

The visual similarity between an image matrix and Xenakis's matrix sparks the possibility of transferring strategies that Xenakis employed to image sonification. We can set the timbral qualities on the vertical axis and use the horizontal axis for durations. Since an average image file allows plenty of data points we can map many of the parameters of instruments. To begin with the top-level organization of the work, we need to reduce the amount of data. Dividing the time axis equally would result in a pixelated version of the picture which does not present the shapes and patterns we observe in the original picture especially when the pixel size is big. It would also lead to changing the values at the same intervals and synchronously. We need to keep the structural quality of the image while dividing it into sections at different detail levels. The desired process can be illustrated as follows: If we look at a picture from a distance we experience large sections of colors and shapes. When we get closer and closer to the painting new details are revealed at each step until we can see the brushstrokes of the painter, the cracks in the paint layers, subtle color gradients, etc.

The research in computer vision provides tools for dealing with difficult tasks. Superpixel algorithms can group pixels according to similarities in color, brightness, and other low-level properties at a very high success rate. (Ren & Malik 2003) Superpixels are powerful to retain the overall look of an image while diminishing data. The size of the superpixels and the number of superpixels that can be found in an image is inversely proportional. If we divide an image into a higher number of sections we will lose more details but we obtain a more general view that we can use for parameters we demand to change less often. We opted to use SLIC (simple linear iterative clustering) superpixel algorithm that utilizes a k-means clustering, a machine learning algorithm. SLIC is simple to use, fast, and successful at detecting boundaries. (Achanta et al. 2012) Speed of the algorithm is important for testing different segmentation strategies. Depending on the arbitrary number we defined for the number of segments, the SLIC algorithm tries its best to identify similar areas. If the chosen number of segments is low then larger areas are selected by the algorithm. Increasing the number produces smaller segments, hence a more detailed picture. Then the next step is averaging the values in each segment. When we draw a line horizontally through the picture we get changing values that we can use for any parameter in our musical design. Thanks to the vast amount of data an image file can provide we can draw as many lines.

Figure 3: Segmentation with SLIC algorithm at different levels

To test our method we've used a picture with 6000 px width and 4000 px height that one of the authors took with the Sony A6000 digital camera. The programming environment was Python which is popular amongst the data science and machine learning community. Using the *scikit-image* library we were able to implement the SLIC segmentation (van der Walt et al. 2014). We decided to use 30, 300, and 3000 segment numbers and the non-segmented version after experimenting with several other options. 30 segments provided the largest sections as expected. The segment areas got smaller as the segment numbers rose. In image segmentation, the extracted data by drawing horizontal lines on the picture was plotted to visualize the transformation for a better understanding of the result of choices with location and segment numbers.
For our purpose of validating the usefulness of the image sonification method, the sound generation algorithm should have been relatively simple yet versatile. We picked granular synthesis for the project since it can generate variable textures with controlling few parameters. Granular synthesis is a method to get the slice of audio preferably less than 100 ms and then combining those sound portions in large amounts to create cloud-like textures (Roads 2001). The sound realization of the piece was accomplished in Csound, the audio programming environment. Csound can work in the orchestra (the instrument definitions) and score (note-events) paradigm (Boulanger 2000) so that we were able to generate a score file in Python as a list of values that describes the changes in instrument parameters. Csound renders the instrument and score definitions as an audio file which can be a short process. This workflow enables us to experiment with several options while keeping the results as Csound score file and audio file for further analysis. The software instrument is based on Csound's *grain*[1] opcode that we chose to map the *grain duration, density, and pitch* parameters since they are the most definitive ones on the sonic quality and the texture. The variability on the sound was achieved using separate *grain* functions for left and right channels while keeping grain, duration, density, and pitch the same but assigning random values for amplitude and pitch offset parameters. The only parameter for the blandly added reverb is the amount of the effect to the dry signal. The instrument accepted the 567 milliseconds of a cello sound as input which was demonstrating a forced bow gesture. Keeping the duration longer resulted in more like a sampler instrument sound against the granular sound which created enticing contrast throughout the piece. The piece was designed to endure 3 minutes for 6000 data points that allow 30 milliseconds for each data point. At each step, values are calculated and written to a text file for rendering in Csound later. The minimum and maximum values for mapping were based on the explicatory needs of the demo piece.

We mapped data onto musical parameters at different time scales with image segmentation. (fig.3 and fig.4) In the course of the piece, parameter values remained the same for the allocated time which resulted in significant formal structures. The generated audio demonstrated that with image sonification it is viable to create a multi-scale form that overcomes the bottom-up approach inherent to sonification and algorithmic music in general. However, we should note that we had to make some design choices to make this experiment easy to process and evaluate the results: The duration of the piece was rather short; A single instrument with a limited number of parameters was used; Many aspects of electroacoustic music had to be dismissed. The proposed method will be more convincing in large-scale compositions with numerous instruments and parameters.

[1] https://csound.com/docs/manual/grain.html

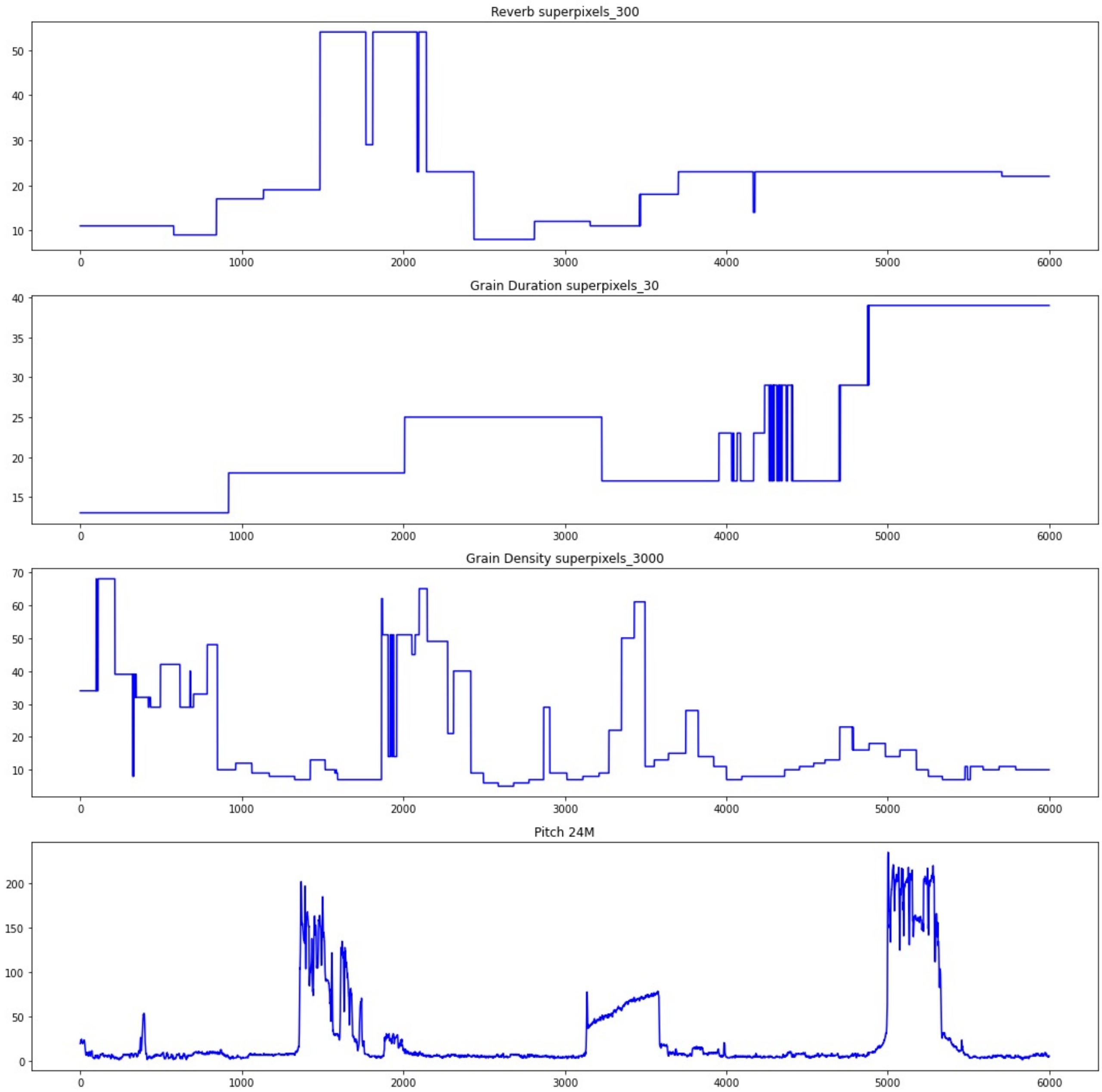

Figure 5: Visualization of extracted data from the image for each parameter and selected segmentation

References

Achanta, R., A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. 2012. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods.” IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11): 2274–82. https://doi.org/10.1109/TPAMI.2012.120.

Arsenault, Linda M. 2002. “Iannis Xenakis’s Achorripsis : The Matrix Game.” Computer Music Journal 26 (1): 58–72. https://doi.org/10.1162/014892602753633542.

Attenborough, Mary. 2003. “Probability and Statistics.” In Mathematics for Electrical Engineering and Computing, 493–532. Elsevier. https://doi.org/10.1016/B978-075065855-3/50047-4.

Boulanger, Richard, ed. 2000. The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming. Cambridge, Mass: MIT Press.

Childs, Edward. 2002. “Achorripsis: A Sonification of Probability Distributions.” In Proceedings of the 2002 International Conference on Auditory Display, Kyoto, Japan, 2002. Kyoto, Japan.

Collins, Nick. 2009. “Musical Form and Algorithmic Composition.” Contemporary Music Review 28 (1): 103–14. https://doi.org/10.1080/07494460802664064.

Jackson, Wallace. 2015. “The Color of Digital Imaging: Color Theory.” In Digital Image Compositing Fundamentals, by Wallace Jackson, 23–29. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4060-1_4.

Kramer, Gregory, Bruce Walker, Terri Bonebright, Perry Cook, John Flowers, Nadine Miner, John Neuhoff, et al. 1999. “The Sonification Report: Status of the Field and Research Agenda. Report Prepared for the National Science Foundation by Members of the International Community for Auditory Display,” January.

Pennebaker, William B., and Joan L. Mitchell. 1992. JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold.

Roads, Curtis. 2001. Microsound. Cambridge, Mass: MIT Press.

———. 2015. Composing Electronic Music: A New Aesthetic. Oxford ; New York: Oxford University Press.

Spiegel, Laurie. 2018. “Thoughts on Composing with Algorithms.” In The Oxford Handbook of Algorithmic Music, edited by Roger T. Dean and Alex McLean. Vol. 1. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190226992.013.26.

Supper, Martin. 2001. “A Few Remarks on Algorithmic Composition.” Computer Music Journal 25 (1): 48–53.

Walt, Stéfan van der, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. 2014. “Scikit-Image: Image Processing in Python.” PeerJ 2 (June): e453. https://doi.org/10.7717/peerj.453.

Xenakis, Iannis. 1992. Formalized Music: Thought and Mathematics in Composition. Translated by Sharon Kanach. Rev. ed. Harmonologia Series 6. Hillsdale, NY: Pendragon Press.

Yeo, Woon Seung, and Jonathan Berger. 2005. “Application of Image Sonification Methods to Music.” In In Proceedings of the International Computer Music Conference.