

TecnoLógicas

Tecno Lógicas

ISSN: 0123-7799

tecnologicas@itm.edu.co

Instituto Tecnológico Metropolitano  
Colombia

Gómez, Daniel

Spectron: graphical model for interacting with timbre

Tecno Lógicas, núm. 22, julio, 2009, pp. 61-74

Instituto Tecnológico Metropolitano

Medellín, Colombia

Available in: <http://www.redalyc.org/articulo.oa?id=344234314005>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System

Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal

Non-profit academic project, developed under the open access initiative

# SPECTRON: GRAPHICAL MODEL FOR INTERACTING WITH TIMBRE

*DANIEL GÓMEZ<sup>1</sup>*

## Resumen

Los algoritmos para crear y manipular el sonido por medios electrónicos o digitales han crecido en cantidad y complejidad desde la creación de los primeros sintetizadores análogos. Sin embargo, las técnicas para visualizar estos modelos de síntesis no han crecido a la par de los sintetizadores hardware o software.

En este artículo se muestran posibilidades para representar y controlar gráficamente el timbre, basadas en la visualización de los parámetros involucrados en su modelo de síntesis. Un grupo de datos muy simple fue extraído de un sintetizador substractivo comercial y analizado con dos aproximaciones diferentes, reducción dimensional y visualización abstracta de datos. Los resultados de estas aproximaciones diferentes fueron usados como lineamientos para crear un prototipo de sintetizador digital: el sintetizador Spectron. Este prototipo usa el gráfico de Amplitud vs. Frecuencia como su principal herramienta para informar a cerca del timbre e interactuar con el, fue desarrollado en PureData y su control plantea una simplificación en la cantidad de variables de un oscilador clásico al mismo tiempo que expande las posibilidades para generar timbres adicionales a los de estos osciladores clásicos.

## Abstract

The algorithms for creating and manipulating sound by electronic or digital means have grown in number and complexity since the

---

1 Production Engineer, Master in Digital Arts. Instituto Tecnológico Metropolitano, Teacher / researcher, danielgomez@itm.edu.co

creation of the first analog synthesizers. The techniques for visualizing these synthesis models have not increasingly grown with synthesizers, neither in hardware nor in software.

In this paper, the possibilities to graphically represent and control timbre are presented, based on displaying the parameters involved in its synthesis model. A very simple data set was extracted from a commercial subtractive synthesizer and analyzed in two different approaches, dimensionality reduction and abstract data visualization. The results of these two different approaches were used as leads to design a synthesizer prototype: the Spectron synthesizer. This prototype uses an Amplitude vs. Frequency graphic as it's main interface to give information about the timbre and to interact with it, it's control offers a simplification in the amount of variables of a classic oscillator and expands its possibilities to generate additional timbre.

### **Keywords**

Data visualization, timbre, Spectron, sound synthesis.

## INTRODUCTION

Sound synthesis is a practice aimed to create and control sound. Each synthesis technique has a model that can be expressed in an algorithm, but in order to be manipulated, the model should have some sort of graphical or physical device that makes the algorithm feasible. There are many ways in which the relation between an algorithm and its graphical or physical device arises.

There are patch languages for synthesis that use an iconography of the various elements that constitute a certain type of synthesis (Loy, 2007) (Roads, 1996), and these can be used to sketch a sound process on paper. These patches, which explain synthesis' elements and data flow, can also be programmed in computer language or created by using electronic circuits, making possible new ways of representing sound generation equations: a blueprint for the construction of a circuit, the circuit itself or the code used in a certain computer language to create a synthesis.

These written patches, algorithms, electronic blueprints or software codes are all representations of a model of synthesis. However, these can be only decoded by experts: programmers, electronic or sound engineers. Those systems and their data are not useful for the general public; (Shedroff, 1994) it is mandatory an interface that allows the tweaking of the variables involved in the algorithm, the system, or the box, to make the use of synthesis possible.

If we observe these synthesis interfaces carefully, we manage to see those as a final representation of synthesis. In the first days of analog synthesizers, a wooden panel full of knobs and connections became the user's physical and visual representation of sound synthesis. (Pinch, Trocco, 2002)

That panel is the top of the iceberg of the engineering perspective inside of synthesis algorithms, and it is still closer to the expert side than to the user side; additionally, it recalls the problem described by Shedroff (1994), that "too often we deluge our audience with data instead of information, leaving them to sort it out and make sense of it".

The main reason for the lack of clarity may reside in the fact that the sound model and its implied data flow are not understandable at sight and touch: all the variables have the same “face”, which is a black knob with no differentiation. So the first encounters with a synthesizer (be it software or hardware) are an attempt to make sense of all the knobs and buttons present in its interface. Besides, the relation between these buttons and the sound generated are mediated by this synthesis model, intended only for qualified people and not for the common user.

Thus, to truly understand the way the tweaking of knobs truly affects timbre, the user must refer to the math and physics of sound. Finally, timbre becomes an aspect conducted by science, which an enthusiastic synthesist must study and which is not explained by the synthesizer itself.

The origin of this multiple layer separation between the sound generated and its control is understandable, as the synthesizer was originally a scientific musical device, a box that came from laboratories intended for studios or performances. The technology inside the box was advanced enough to grasp all the attention of its designers and consequently caused that the efforts were put mostly in circuitry. (Pinch, Trocco, 2002)

One of the main problems with current commercial synthesizer graphic displays (Magnusson, 2005) is that there is no graphical or physical relation to timbre, but there's a relation to the magnitude of variables of an abstract synthesis model. Furthermore, neither the synthesis model, nor the exposed relationship between spectrum and knobs is understood by the arrangement of the knobs on the board. Therefore, the need of creating interfaces that could bring comprehension and control closer to the user marked a field of work exposed below.

## RELEVANT PROJECTS

This perspective of establishing new parameters for the use of sound synthesis has generated some remarkable projects that

analyze multi-dimensional sound-tool control in different and interesting ways.

The techniques implemented by Momeni and Wessel (2003) are a good example of dimensionality reduction applied to diverse sound algorithms. Focusing in the use of digital drawing tablets for controlling, a multi layered Gaussian Space is created, where gaussian bells represent the regular distribution of a certain variable in a two dimensional space. Many bells coexist on the same 2D space thence, a point in that space becomes a vector for many variables.

The work made by IXI systems (Magnusson, 2005) is another approach to the experimental graphic control of sound tools. Various GUI instruments were created, which broke common paradigms on the way interaction with sequencing, spacialization and sound synthesis should be made. IXI projects are very particular sound tools and their interfaces are created specifically for each tool.

Psychosynth (Psychosynth, 2008) is another example of a relevant project. It is a digital synthesizer based on the interface of the reactable\* (Jorda, Kaltenbrunner, Geiger) synthesizer. The reactable\*'s GUI interface is designed for its specific sound engine and as a complement for its physical interface. The relation between the synthesis model proposed in the reactable\* and the physical disposition of its control variables fits perfectly and logically, and therefore make it a complete instrument. Although Psychosynth is not an original work, it is a relevant reference for two-dimensional interfaces for timbre.

Many other projects have been proposed on this field, but they are not cited on this paper because the idea is to expose some general lines of work that differ from each other and that complement the panorama of graphic tools for working with sound. Our confrontation of these different lines of work with our own projects is explained below.

## METHODOLOGY

The first step is to gather data that could be used for experimentation. We select a popular tool such as Steinberg's A1 synthesizer

to extract the values of 45 variables involved in its synthesis process. The data of three preset sounds included in the A1

is extracted: a bass sound, a pad sound and a percussive sound. All the 45 variables are tabulated and set out to be exposed to geometric representation.

### Dimensionality Reduction

Data is exposed to the light of dimensionality reduction techniques, with the expectation of representing all the 45 variables in a bi-dimensional or tri-dimensional space. There are many techniques that prove to be useful in some specific cases (Van Der Maaten, Postma, Van Den Herik, 2007) but most of them are cases of a  $\mathbf{D} \times \mathbf{n}$  matrix, with  $\mathbf{D}$  being the number of variables and  $\mathbf{n}$  the number of data vectors. The variable  $\mathbf{n}$  is generally a large and unknown number; different from the dataset of a given sound program in a synthesizer where  $\mathbf{n}$  is 1. These techniques seem to be useful for different data sets, although they could be useful in experiments, by using  $\mathbf{n}$  vectors as each preset sound of a synthesizer of  $\mathbf{D}$  variables.

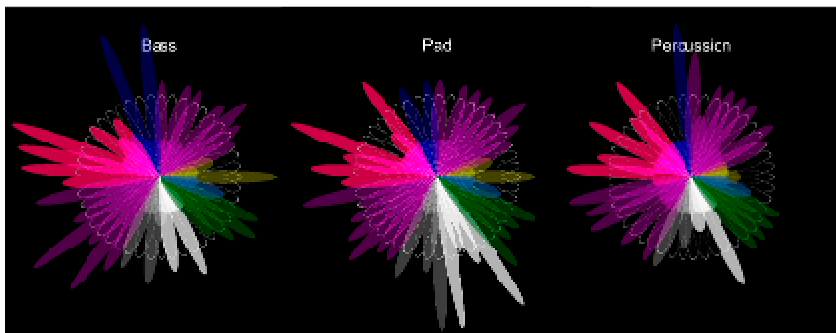
In addition, these are tools meant primarily for data visualization, more than for interaction. Nevertheless, some researches have had, as mentioned before, successful results following a dimensionality reduction approach (Momeni and Wessel, 2003).

### Abstract data visualization

The results of using abstract geometry techniques to control the 45 dimensional mathematical expressions, have led us towards particular geometries with very complex graphic elements. A patch in GEM and PureData (Puckette, 1996) was created to use the dataset to generate a graphic for each instrument variables. As seen in figure 1, these graphics, although stimulating for the eye, proved misleading for comprehension of a single sound, but useful for the comparison between multiple timbre patterns. There are

many examples of this kind of problem when dealing with abstract visualization of high dimensional data. (Tufte, 1990)

Perhaps a hierarchical classification of the variables of the model (for example depending on how extreme or subtle is their impact on timbre) could be a next step to continue this abstract geometric exploration for further projects.



**FIGURE 1** EXAMPLE OF ONE OF THE DATA VISUALIZATION TECHNIQUES USING GEM FOR THREE SOUND PATCHES: BASS, PAD AND PERCUSSION.

### Amplitude vs. Frequency

As the first goal was to obtain a tool that could be used with general information available rather than creating a graphical language for a specific purpose, the work led to search for a common visual reference towards timbre. The central images of sound found in wave physics (Loy, 2007), synthesizer manuals (Novation, 2001) (Roland Corporation, 1994) (Blanchet, 2004), and sound analysis (Roads, 1996) are the Amplitude vs. Time (A vs. T), and Amplitude vs. Timbre graphics (A vs. F). A specific focus on A vs. F was taken, avoiding the waveform approach used in most of additive synthesizers. This graph is a visual representation of timbre and the goal of sound synthesis is to create desired timbres.

The first result in the exploration of the use of the A vs. F graphic as a central interface for the control of timbre is the Spectron Synthesizer prototype.



## SPECTRON PROJECT

This work focuses on an interactive representation of the spectrum and the idea of the A vs. F graph as an obvious image of the result of patching or synthesizing a sound. This tool focuses the user's work directly on the spectrum, and automatically takes charge of the waveform and algorithm tweaking.

### Spectron Synthesizer

The first activity is to develop diverse types of synthesis models in PureData, as complete and full of features as commercial plug-in synthesizers. Therefore, Additive/subtractive, FM and granular synthesizers are created.

The goal of this first prototype is to create a tool that shortens the distance between user interaction and the sound generated with basic additive synthesis. The resultant spectrum of this synthesis process is divided into three small spectrum fragments called "Spectrons," that (in the way a simple additive synthesizer adds oscillators) create the main output. That is: each Spectron is added with each other so that the final spectrum is the resultant of the addition of them all.

### Spectron

As seen in Figure 2, the Spectron control has 4 variables: Amplitude, First Harmonic, Higher Harmonics angle slope level, and Higher Harmonics Interval.

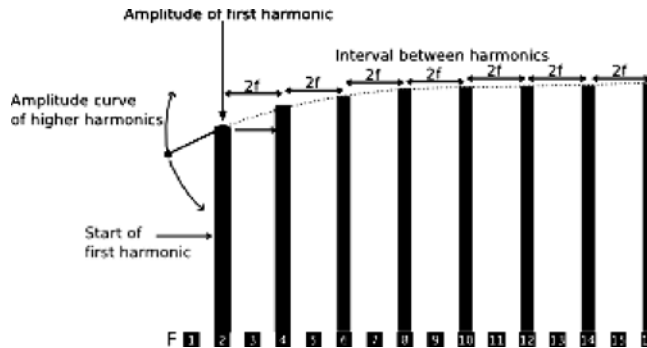


FIGURE 2 SPECTRON DIAGRAM WITH 4 VARIABLES.

Amplitude: level of the first harmonic. It is the reference for controlling the slope.

Angle: the inclination of the slope that the higher harmonics will follow to get their amplitude so, it is the way to control the presence of high tones in the resulting spectrum. The equation used to control the amplitude of the **nth** harmonics is:

$$A_n = (\text{Amplitude}/n) + (\text{angle} \times \text{amplitude} \times n - 1/n) \quad (1)$$

where, **n** is the index of the harmonic.

By tweaking the angle control up and down, we emphasize the higher harmonics of the Spectron so they can be much higher than the first harmonic, yielding to bell-like sounds.

First harmonic: the multiple of **F** that denotes the start of the first harmonic of the Spectron. You can switch it to higher multiples of **F**.

Interval between harmonics: the equidistant interval that separates the high harmonics in multiples of **F**.

The use of these variables is a recursive way for manipulation, but both the equation of the angle and the Interval are functions that can be programmed in many different ways.

The Spectron and its construction methods have the particularity of being able to act as classic oscillators creating classic wave

types but at the same time, moving towards complex waveforms and spectrum.

### **Spectron Variable Analysis**

A classic oscillator found in an analog or virtual synthesizer has 5 basic variables: Amplitude, Wave Type, Finetune, Pulse Width, and Octave. That makes 4 continuous variables and one discrete variable. The Spectron has 4 variables: 2 continuous variables and 2 discrete variables. With a less number of variables, a Spectron can generate the classic waveforms (except for noise) and can also transform them into more complex waveforms, not by addition of Spectrons, but by the change in higher harmonic composition.

Modulation is available to control the amplitude and the micro tuning of each Spectron yielding micro and mid time complexity.

### **Spectron Interface**

On top of figure 3, the resulting spectrum can be seen as the addition of three Spectrons, each one distinguished by a different color. Below the spectrum we have the corresponding waveform. The color code that differentiates each Spectron is useful to determine the amount of energy each one is adding to a particular harmonic. This color code continues its way below, to a menu bar where you can select either a Spectron or the modulation panel for editing.

Clicking on any of the three colors of the resulting spectrum, makes the lower portion of the interface show the Spectron associated with each color.

Each Spectron is complemented with its own waveform, just for informational purposes.

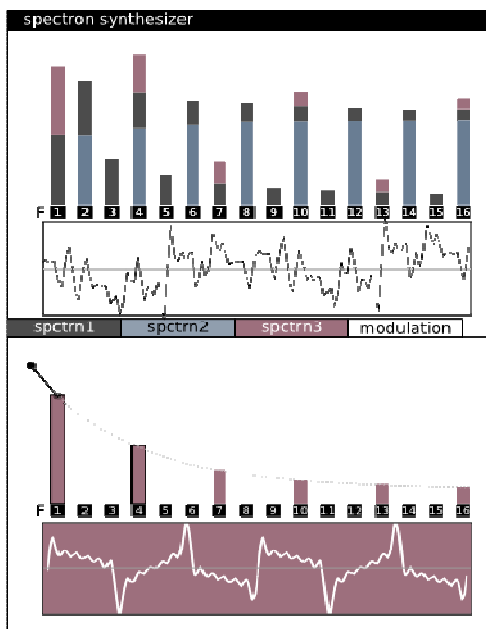


FIGURE 3 SPECTRON SYNTHESIZER GUI

## DISCUSSION

The algorithm used in the Spectron prototype to create and control higher harmonics in each Spectron is not complete yet, and many other ways can be implemented, apart from the one exposed on this paper. Nevertheless, there has been an improvement in the amount of variables used to control a basic oscillator that minimizes the number of variables and amplifies sonic possibilities.

Other techniques, such as FM synthesis and granular synthesis can be attached to the prototype so that, according to the spectrum manipulation on the interface, an algorithm decides what the simplest way to create a particular spectrum distribution is. This could be useful in diminishing the DSP consumption of the algorithm.

The last argument helps us assume the fact that digital processing consumption of elaborate graphics is also a very

important aspect that should result in sound tools focusing more on sound than image, since graphics are kept as simple as possible in plug-in synthesizers. This aspect eases the use of other plug-ins, but restricts the use of these tools to experts who understand the model of the specific synthesizer and the values of variables that should be input to the interface ahead before achieving the desired sound.

There is still a wide path to be explored in the abstract representation of sound algorithms; due to its use and constant revision, this could lead to a new standard of visualization, in the same way as the wooden panel with knobs became a standard. Also, the use of Dimensionality Reduction tools to analyze the problem has yet to be developed in ways that transcend graphical comprehension and lead to physical interaction.

## CONCLUSIONS

Projecting specific solutions is still the most effective way to set about the control of digital sound tools. A general solution that could analyze all the types of situations is still not in sight. Nevertheless, sound synthesis techniques are not that abundant, and a systematic study for each model and the relations between its variables towards interactive visualization could lead to cleaner and friendlier tools.

The data sets generated by the variables of a synthesis algorithm (or vectors as called above) are promising material to develop visualization strategies focused on: sound comparison, the creation of maps that explain the data flow in an algorithm, the relationship between its variables and finally the design of informative tools for the use of digital synthesizers.

Although the A vs. F graphic is a general tool for visualizing spectrum, many other approaches can be taken than the presented in this paper, especially if those approaches have an emphasis in making some kind of visualization to the model of the synthesis algorithm.

## REFERENCES

- Loy, G.. (2007). *Musimathics, The mathematical foundations of music*. Cambridge, Ma: MIT press. Pp. 365 - 383
- Shedroff, N. (2008). *Information Interaction Design: A unified Field Theory of Design*. <http://www.nathan.com/thoughts/unified/>, accessed December.
- Pinch, T. & Frank T. (2002). *Analog Days, The invention and impact of the moog synthesizer*. Cambridge, Ma: Harvard University Press.
- Wessel, D. and Momeni, A. (2003). "Characterizing and Controlling Musical Material Intuitively with Geometric Models." *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, Montreal, Canada, may 22 - 24.
- Van Der Maaten, L.J.P., Postma, E.O., Van Den Herik, H.J. (2007). *Dimensionality Reduction: A Comparative Review*. [Http://tsamfich.wdfiles.com/localfiles/apuntes/TPAMI\\_Paper.pdf](http://tsamfich.wdfiles.com/localfiles/apuntes/TPAMI_Paper.pdf)
- Puckette, M. (1996). *Pure data: another integrated computer music environment*. *Proceedings the Second Intercollege Computer Music Concerts*, Tachikawa, Japan.
- Novation. ( 2001). *Nova Owners Manual*.
- Roland Corporation. (1984). *Juno-01 Owner Manual*.
- Roads, Curtis. (1996). *Computer Music Tutorial*. Cambridge: MIT press. Pp 85 - 114
- Jorda, Sergi. (2005). "Digital Lutherie, Crafting musical computers for new musics' performance and improvisation." Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain.
- Tufte, Edward. (1990). *Envisioning Information*. Cheshire, Ct: Graphics Press.
- Magnusson, T. (2005). *ixi software: The Interface as Instrument*. *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*, Vancouver, BC, Canada.
- Blanchet, Jean-Michel et al. (2004). *Moog modular v2 User's manual*. Arturia Sarl.
- Psychosynth. <http://www.psychosynth.com/doku.php> accessed December 2008

- Jorda, S. & Kaltenbrunner, M. & Geiger, G. & Bencina, R. (2005). The Re-actable. Proceedings of the international coomputer music conference ICMC, Barcelona, Spain.