PixelAudio:
A toolkit for intermedia composition

Paul Hertz
Independent artist
Chicago, Illinois, USA
ignotus@gmail.com

PixelAudio is a Java software library for the Processing development environment. It offers a toolkit for intermedia composition where images and audio signals are mapped onto each other. Features include curve drawing to activate audio sampling and additive audio synthesis for simultaneous sound and pattern generation. PixelAudio builds on historical experiments with cross-modal sensory instruments such as colour organs and lumia, and more recent experiments in digital transcoding, glitch, and intermedia experiences. The library tools fill out a theoretical niche in musical/visual composition that sits above simple data equivalence ("bytes are bytes") and advances towards the interplay of compositional structures that can be passed back and forth between different media.

Intermedia. Software. Music. Image. Animation. Composition.

1. INTRODUCTION: INTERMEDIA

In a landmark essay in the first newsletter of the Something Else Press, Fluxus artist, theorist and publisher Dick Higgins describes *intermedia* as an art practice that crosses the boundaries between media or fuses media (Higgins 1966). What does the fusion of media imply? He clarifies what he means mostly by example. From his diagram of intermedia artforms current at the time he wrote, from the two following issues of the newsletter and other publications from Something Else Press, we may imagine that intermedia implies something more than a multimedia opera and something less than a synesthetic experience. At one point he suggests that "structural elements" are bound up in the fusion of media. Thus, intermedia is neither a total art work nor a recreation of sensory simultaneity, but a blending of different artforms. Higgins also suggests that intermedia is an "inevitable and irreversible historical innovation" rather than an artistic movement, such as Romanticism. On the threshold of the electronic age, with the critique of avant-garde authority erupting into full postmodern chorus, he views the rupture of categories as essential to art as a vital experience. He proposes intermedia as a response to the exhaustion of "the concept of the pure medium" that develops when art is reified into marketable objects. He uses it in the plural: "Intermedia differ from mixed media in that they represent a fusion conceptually of the elements." (Higgins 1976) We find further clues in *A Short History of Pattern Poetry*, where he describes the fusion of poetry and geometric patterns as an historical and cross-cultural blend of media (Higgins 1987). He expects intermedia, in time, to become media and spawn further hybrids.

Following his lead, we may explore intermedia as an historical practice occurring in many cultures. For example, in the European cultural tradition, colour organs and visual music have a long if sporadic history, composing colour and shape in much the same way a musician composes pitch and duration. Fusion of the senses inspires Baudelaire's poetics of correspondences and the Symbolist movement that followed him: "Nature is a temple of living pillars…where perfumes, sounds and colours correspond." (Baudelaire 1857) He conflates sensory fusion with transcendental insight, possibly reflecting upon the new mathematics of wave functions: fusion of the senses evokes the perception of the "harmonic vibrations" inherent in all natural phenomena (Leroux 1831, Hertz 2012). Even earlier, in ritual and in esoteric systems of knowledge, such as those propagated by Ramon Llull, Robert Fludd, or Athanasius Kircher, *ars combinatoria* acts as the structural principal linking cosmology, musical scales, numbers, concepts and symbols (Zielinski 2006, Zweig 1997). Similar systems can be found in the I-Ching of ancient China or the West African Ifá, both of which use binary numbers to encode cultural knowledge. These systems often appear in the guise of magic or divination, cueing mystical insights within a formal order. Janet Zweig suggests that formal orders emerge as “symbolic logic, semantic invention, or pure process and play,” a precursor to both scientific and artistic inquiry (Zweig 1997). For example, Kandinsky’s *Point and Line to Plan*e and Klee’s *Pedagogical Sketchbook* both propose the formal order of music as a paradigm for abstract art.

In his influential book *Expanded Cinema,* Gene Youngblood speaks of “synaesthetic cinema” replacing exposition, the mode of traditional narrative cinema, with “evocation … primarily poetic in structure and effect” (Youngblood 1970, p. 92). In contrast to the poetic, hybrid, or syncretic poetics that Higgins and Youngblood espouse, experimental filmmaker John Whitney Sr. foresees in computer animation “a visual domain of harmonic relations to match that domain of music” made possible because “computer graphics is the only instrumentality whereby the harmonic essentials can be known and made visible and manipulated” (Whitney 1973). Whitney’s book *Digital Harmony* provides both a theoretical and algorithmic foundation for his computer animations, many of them based on the fundamental harmonic proportions that were familiar to Pythagoras (Whitney 1980).

With Youngblood and Whitney we seem to arrive at two distinct approaches to intermedia practice: a poetics grounded in language and meaning and a compositional theory grounded in numbers and abstract form. We can see these different approaches in the work of contemporary artists such as Jack Ox, concerned with conceptual blending and language, and Vibeke Sorensen, concerned with algorithmic continuity across different media (Ox 2015, Sorensen & Steiger 1989). The fact that both artists view music as central to their practice shows how either approach can function well as a basis for making art. Indeed, we might ask whether the most critical function of intermedia is to simply to enable the creation of new artforms and new works. Metaphors and shared structures may fall away like scaffolding once an intermedia work is realized—do they remain as generative forces even if outside immediate perception? We might further ask—if metaphors and structures enable the same outcomes, are they fundamentally different? If they aren’t, wherein lies their similarity?

2. INTERMEDIA STRATEGIES

Before we can answer this question, we have to expand our concept of metaphor and deconstruct our process of formal composition. Fortunately, there are theorists to guide us. Lakoff and Johnson, in *Metaphors We Live By*, argue that human thought processes are largely metaphorical, if by metaphor we understand conceptual structures that order thought and give rise to the more familiar linguistic metaphors (Lakoff & Johnson 1980). Metaphors involve partial mappings from one domain of experience to another and arise both from bodily experience and from cultural acquisition. For example, we think of time as space or as motion (ahead, behind, flying), imagine our visual field as a container (out of sight), imagine virtue, status and benefit as a direction (up). The apparent banality of such examples demonstrates how we tend to pass them over, grasping meaning without attending to metaphor. Some twenty years later, Fauconnier and Turner developed metaphor theory further, into “conceptual blending,” a process of mapping different experiential and linguistic domains onto one another to derive a new, blended domain, with attributes of both precursors expanded or muted to construct new signifiers (Fauconnier & Turner 2003). Their theory benefits from validation by psychologists “whose methodologies included both clinical experimentation and computer modelling” of analogy and mental images (ibid. p. 33). Critically, they view the “complex emergent dynamics” of the imaginative mind as the fundamental source of meaning in language and experience, distinct from the forms that combine to reach “conceptual integration” (ibid. p. 22). Meaning is not another structure hidden within or inscribed upon the structures within the mind’s field of attention, but the very process that unifies them.

In their subsection “Crazy Numbers,” Fauconnier and Turner discuss imaginary numbers. “Imaginary” numbers such as the square root of -1 are of course entirely real as mathematical symbols and tremendously useful—one can hardly imagine doing 3D graphics without their three-dimensional analogue, the quaternions—nevertheless, mathematicians once regarded them as impossible. Fauconnier and Turner point out that complex numbers, consisting of a real and an imaginary part, became widely accepted once they were understood as a blending of numbers and spatial vectors within the complex plane. They argue that mathematics as a whole depends on similar conceptual blending. In so far as blending is a function of bodily and social experience, their argument squares with embodied mind and social production theories of mathematics and runs counter to formalist and realist theories, for which mathematics is grounded in string manipulation rules or inherent structures in the world. As with their argument regarding meaning, we seem to enter into the realm of philosophical discourse more than that of empirical proof, though time may tell.

I suggested some years back that the concepts behind “synesthetic art” were analogous to imaginary numbers, mostly valuable as a stimulus to new work (Hertz 1999). Am I suggesting the same thing about intermedia, that it is just conceptual scaffolding? Rather, I would say that intermedia explicitly and implicitly functions through conceptual blending. In the compositional process blending is an explicit generative strategy, akin to conceptual blending in mathematics but more definably so: the elements and operators in intermedia composition are tied to experiential parameters that emerge in media productions. They function as an *a priori* analogue of the hidden processes that Lakoff and Johnson attribute to metaphor and Fauconnier and Turner attribute to conceptual integration. People who experience an intermedia production may feel the trace of the underlying process, the sense of unity, without awareness of the complexity beneath. This is, apparently, how our mind works most of the time. In other intermedia productions, the blending is overt, as in poetic metaphor. It is not meant to dwell in the subsoil, like a fungal network, but to catch our attention and call upon us to blend its elements consciously. In most intermedia art, both processes are at work.

Intermedia composition takes place in a continuum of distributed blending operations, before and after the experience of intermedia art, within a network engaging artist and audience in analogous but separable processes. The complex symbolic operations of composition, whether mathematical, algorithmic, or analytical, operate in much the same way that harmonic progression does in Western music. Most people are not aware of it, but it is essential to creating a song or a sonata, and people perceive its unifying influence. Apparently simple events that clamour for awareness—a cadence, a moment of mirror symmetry, a pirouette—rely on perception, but they are no less complex. Such events call for audience complicity without necessarily attaining immediate awareness. The intermedia composer relies on the vast apparatus of human cognition, of which we are necessarily unaware, to blend events.

In answer to my questions, then: Intermedia composition may operate as a structural framework prior to the unfolding of an intermedia event or as conceptual integration in the wake of an event, but in both situations it relies on the integrative power of the imagination. The metaphors and structures involved in composition and reception are not fundamentally different, but they occur in different stages, times and levels of attention in the social process of intermedia art. The asymmetries of creation and reception, the certainty that the work will be transformed and propagated by the imaginative power of its audience even more than by the skill or determination of the artist, may not ensure the work's longevity, but they do bind it to the continual, shared task of engaging the imagination.

3. PIXELAUDIO

PixelAudio, a Java software library I have created to aid in intermedia composition, operates with algorithmic structures rather than with poetic metaphors (Hertz 2025). Its core concept is *mapping*, in this case between audio and image data. As a strategy for composing and for software design, mapping is distinct from *transcoding*, where data from one media representation is inscribed directly into another, and from *structural interchange*, where the high level organizing structures of one medium are repurposed within another medium.

3.1 Mapping as a Compositional Process

As should be clear from the opening discussion of intermedia composition, mapping is an essential aspect of intermedia and of the theoretical frameworks that support it. These include art practices such as Fluxus performances and happenings, generative processes for media fusion, theories of metaphor and conceptual blending, and the continuum from intermedia author to intermedia audience. Some of the space-filling curves that I use for mapping are fractals, for which there is extensive mathematical literature relevant to art and science, starting with Mandelbrot's *The Fractal Geometry of Nature*. As an internet search for "Hilbert curves and scientific visualisation" will reveal, Hilbert curves are extensively used for data visualisation to reveal regularities that are difficult to observe in one-dimensional data streams. The scientific application of Hilbert curves influenced my own choice to explore them as pattern-making tools for visualising audio signals.

PixelAudio mediates between *images* and *signals.* It is a toolkit for mapping image data objects and audio data objects onto each other. I have purposely kept its codebase rather slim, as though it were cottage-crafted code: thus, PixelAudio may be regarded as an artisanal artwork and a critical statement. That said, I hope to see it used by people other than myself, for purposes I have not imagined. I hope I have made its code and comments clear and cogent enough to make that possible. Its usefulness to my own artistic practice is evident.

3.2 PixelAudio Library Features

PixelAudio maps one-dimensional floating point arrays of audio samples (signals) onto two-dimensional integer arrays of RGB pixel values (images) by means of one-dimensional space-filling curves such as a zigzag or a Hilbert curve. The curve, which we may call the *signal path*, visits every pixel in the image exactly once. There's also an *image path*, in standard row major (left-to-right, top-to-bottom) order for bitmap data. PixelAudio uses lookup tables (LUTs) to map the signal path and the image path onto each other. With a LUT, we can easily locate an audio sample from a given pixel or access a sequence of pixels from a sequence of audio samples.

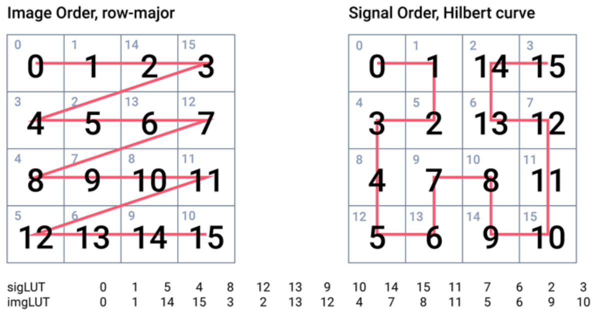

Figure 1: *Two 4 x 4 arrays traversed in row major and Hilbert curve order, with the resulting lookup tables.*

The data format of audio and image are different, of course. Colour is encoded in RGB or RGBA format and audio is encoded as 16-bit floating point values representing amplitude over the interval (-1.0, 1.0). PixelAudio provides various methods to transcode from one to the other. We can, for example, transcode audio data to brightness values, or transcode one or more channels of image colour data to audio, though the result may be noisy. We can save the visualisation of an audio file as an image and later on open it and play it again as audio—as long as the signal path is the same in both instances. We can apply more complex processes, such as a Fast Fourier Transform (FFT), to audio and image in parallel. By reading image data along the signal path, we obtain a one-dimensional array that can be transcoded, processed, and then transcoded back to pixel data. We can process an audio array and write the resulting values along the signal path to an RGB or HSB colour channel. We can also keep audio and image data separate even though we apply the same processes to them, thus maintaining the resolution and quality of both audio and image.

The audio signal array, the bitmap data array, the signal path, the image path and the LUTs are all in one-to-one correspondence. Mapping and transcoding in PixelAudio are implemented by the **PixelAudioMapper** class and the **PixelMapGen** class and its subclasses, which implement different space-filling curves. A PixelAudio application starts by creating a PixelMapGen and plugging it into a PixelAudioMapper instance. PixelMapGen requires width and height parameters to generate the coordinates of the signal path and the LUTs; the image path is built-in to the bitmap. Width and height values determine the length of the signal and image arrays and the LUTs. PixelAudioMapper does not instantiate the signal or image data, but it does provide a wealth of methods for mapping and transcoding arrays of RGB and floating point data.

It's up to the programmer using PixelAudio to create images and audio signals of the appropriate size. PixelAudio does include error handling code and accommodations for images and audio files or buffers that do not conform to the sizes initialized in PixelMapGen and PixelAudioMapper.

Other classes include:

- **WaveData** and **WaveSynth**, for creating animations and audio using additive audio synthesis;
- **Argosy** and its support class **Lindenmayer** (an L-system generator), for creating numerical patterns that can be used to create image patterns or audio rhythms;
- The **pixelaudio.curves** package for freehand drawing to the screen, with point reduction, conversion to Bézier curves and simulated brushstrokes. Affine transforms are built into the various curve classes.

3.3 Working with PixelAudio

Some of the sample code that comes with the PixelAudio library demonstrates the fundamental concepts of mapping and LUTs. Other examples demonstrate possible applications, for example:

3.3.1 Audio as image

An audio signal may be acquired from a file or generated in code. The floating point values over the interval (-1.0, 1.0) in the signal can be transformed to RGB integer values over (0, 255) in various ways. The most obvious is simply to map the audio values to the R, G, and B colour channels. This results in a grayscale image, and one that does not take into account the relative brightness of the different colour channels. An improved method moves the pixel data into the HSB (hue, saturation, brightness) colour space and then uses the brightness channel to represent the audio values. Using the HSB colour space again, the RGB representation can be translated back to audio with sufficient accuracy to be heard as the original audio data. It will be noisy, since in the process we've gone from a16-bit floating point to an 8-bit integer representation of audio samples. Note that the transcoded data follows the signal path. The image is a visualisation of the audio signal that can reveal patterns not evident in the standard one-dimensional timeline view.

3.2.2 Image as audio

If we can transcode audio to RGB and back again, of course we transcode the RGB pixel data in any image to audio samples. For example, we can use the brightness channel of the HSB colour space, mapping from integer data in the range (0, 255) to floating point data in the range (-1.0, 1.0). We should read the image data along the signal path, not in row major pixel order. The kind of curve we use for the signal path will influence the resulting sound of the audio file. In general, though, large scale elements in the image result in low frequency audio components, and small details result in high frequency components. This is apparent even if our image source is a photo: fluffy clouds make low frequency noise, tangled branches make high frequency noise. If our image is generated by the WaveSynth or Argosy classes, we can control the frequency components with precision.

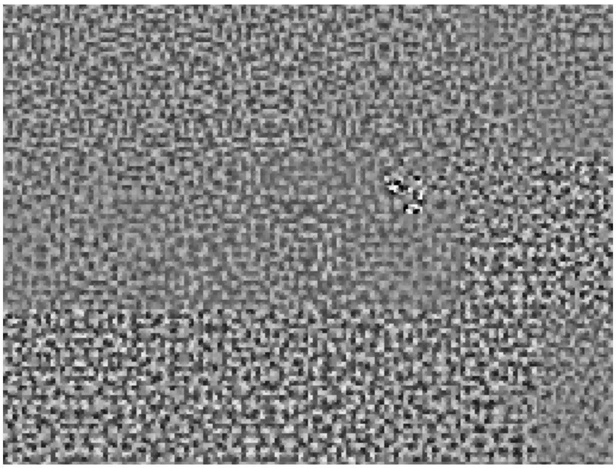

Figure 2: *Audio sample of pitched crotales transcoded to RGB and mapped to a Hilbert curve. Contrast corresponds to loudness. Patterns emerge from different frequencies, which map to the size of blocks of pixels.*

3.2.3 Parallel processing of audio and image

Blurring an image is similar to running a low pass filter on an audio signal, reducing high frequencies. Sharpening increases detail and so behaves like a high pass filter. These effects can be heard when the pixel data is written to the audio signal or vice versa. Since image data and audio data can both be treated as signals, techniques such as Fast Fourier transforms (FFTs) can be applied to both separately. We can apply the same parameters for the transform to audio samples and to pixel values, to simultaneously hear and visualize processes. One of the principal goals of PixelAudio was to apply the same processes to sounds and images without transcoding. A process and its parameters become compositional elements to play with in intermedia composition.

3.2.4 GUI design and interaction

Because there is a precise one-to-one correspondence between pixel location in an image and sample location in an audio signal, the image is in effect a control surface for the signal. The image may be a representation of the data in the signal or it may be any image you care to use. A click on the image can trigger an audio sampling instrument that plays back the signal from an exact location. Dragging the mouse over the image can produce curves and calligraphic brushstrokes (using curve modelling classes in PixelAudio) while simultaneously generating a train of samples similar to granular synthesis. The ordering of the samples within the audio signal will depend on factors such as the geometry of the signal path, the order and number of curve segments, and potentially the recording of time series from gestures across the display screen.

A subclass of PixelMapGen, **MultiGen**, lets you link different PixelMapGen instances to create a single signal path with a custom geometry—for example, one where audio matches words in an image filled with text. In *DeadBodyWorkFlow*, first performed in 2024 at Experimental Sound Studio, Chicago, an image of text in a minimalist typeface spelled out the words of a spoken word performance in such a way that text and recorded audio occupied the same position (Hertz 2024). Drawing on the image activated the recorded text to create layered voices. The sample application **AriaDemoApp** implements the drawing interface and several other methods for use in live performance with PixelAudio.

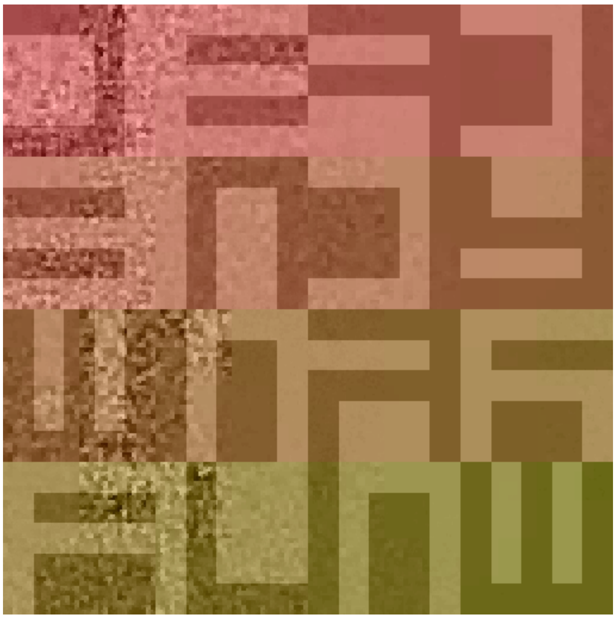

Figure 3: *Overlaid images from* DeadBodyWorkFlow: *text is displayed as geometric letters, audio is the fine granular pattern, and colour follows the signal path*.

3.2.5 Colour organ design with WaveSynth

A WaveSynth object works like an additive audio synthesizer where each sine wave operator controls a colour and the colours are summed to create a pixel value. The frequencies and phases of the operators create patterns in the resulting image. Cyclic shifting of operator phases animates the pattern. I used the WaveSynth and Argosy algorithms to create the animation for the architectural video and audio installation *Campos | Temporales* (Hertz & Walczak 2022). The sample application **WaveSynthEditor** provides a GUI for editing WaveSynth attributes, which it can write to JSON files for subsequent loading and editing.

3.2.6 Scaling Up: Compositional Structures
The mapping, data translation, interaction and additive synthesis features of PixelAudio all operate on individual RGB pixel or audio samples. What about larger compositional structures? Drawing on the image is one of them, of course, and the WaveSynth class with JSON storage lets you create instrumental sounds that are also coloured patterns. The **Lindenmayer** class provides classic string rewriting methods, or "L-systems", developed by Aristide Lindenmayer (Prusinkiewicz & Lindenmayer 1990) for modelling plant growth. The **Argosy** class provides methods for interpreting L-system productions as rhythmic audio events or animated patterns. The generality of L-systems as strings of abstract symbols provides them with a high degree of mobility between media when one is composing an intermedia production. Indeed, one can regard the generator as the critical structure. In *Campos | Temporales*, composer Christopher Walczak used rhythmic patterns generated by a simple L-system while I used the same generator to produce animated geometric figures moving along a Hilbert curve. Musical and animation events might coincide in time, but their compositional binding did not arise from synchronization but from a common generator (Hertz & Walczak 2022).

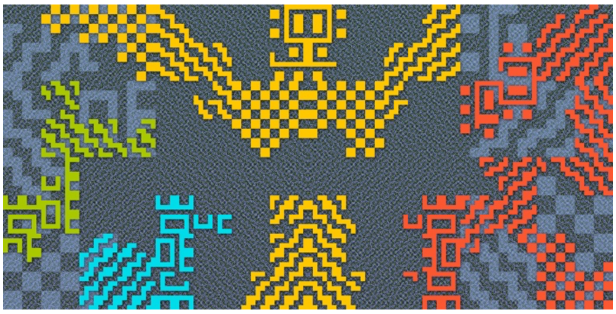

Figure 4: *Still frame from* Campos | Temporales*.*

3.2.7 UDP and OSC implementation
PixelAudio provides basic methods for UDP protocol for networked communications. OSC sits on top of UDP, but is not always necessary. AriaDemoApp provides a UDP implementation using Max, a popular application for audio and video generation. Christopher Walczak and I used the UDP features with Max to perform his composition *Aria and Refrains* at Experimental Sound Studio in Chicago. Many other applications, including the freeware Pure Data (Pd) application, implement UDP and OSC. At the moment of writing this essay, I am experimenting with offloading audio tasks to Max while using PixelAudio for its imaging and control surface capabilities.

4. DISTRIBUTION

As of March 2025, PixelAudio is in late beta development and being tested in workshops. I hope to release version 1.0 in early May 2025, as open source software. An earlier library of mine is part of the Processing contributed libraries distribution. I have hopes that PixelAudio will also be distributed through the Processing website. In any case, the most current release can always be found on my Github pages, linked in the references.

Intermedia was Dick Higgins's response to the reification of art. PixelAudio is released as open source code. Open source operates within a gift economy. Like intermedia, open source art may be "an art form appropriate to artists who felt that there were no boundaries between art and life" (Friedman 2005). I certainly hope PixelAudio will cross some boundaries, from my life to other lives, and along the way stimulate the creation of new works of art.

5. REFERENCES

Baudelaire, C. (1857) "Correspondances," in *Fleurs de Mal*. Poulet-Malassis et de Broise, Paris.

Fauconnier, G. and Turner, M. (2003) The Way We Think; Conceptual Blending and the Mind's Hidden Complexities. Basic Books, New York.

Friedman, K. (2005) The Wealth and Poverty of Networks. In *At a Distance: Precursors to Art and Activism,* MIT Press, Cambridge, Massachusetts.

Lakoff, G. and Johnson, M. (1980) Metaphors We Live By, University of Chicago Press, Chicago.

Hertz, Paul and Walczak, C. Campos | Temporales. https://paulhertz.net/projects/Campos%20|%20Temporales (retrieved 12 March 2025)

Hertz, Paul, PixelAudio (software library) (2025). https://github.com/Ignotus-mago/PixelAudio (retrieved 14 March 2025)

Hertz, P. (2022) IgnoTheory: A Compositional System for Intermedia Art Based on Tiling Patterns and Labelled Graphs. In Darvas, G. (eds.). *Complex Symmetries,* Birkhäuser, Cham, Switzerland. https://doi.org/10.1007/978-3-030-88059-0_3

Hertz, P., Dembski, S. (2018) Fools Paradise: Intermedia in VR. *xCoAx 2018*, Madrid, Spain, 12–13 July 2018, 194–198. Available from: https://2018.xcoax.org/pdf/xCoAx2018-Hertz.pdf (retrieved 12 March 2025)

Hertz, P. (2012) Fischinger Misconstrued: Visual music does not equal synesthesia. In Keefer, C., Guldemond, J. (eds.). *Oskar Fischinger (1900-1967): Experiments in Cinematic Abstraction.* EYE Film Museum, Amsterdam, and Center for Visual Music, Los Angeles.

Hertz, P. (1999) Synesthetic Art: An Imaginary Number? Leonardo, Vol. 32, No. 5, pp. 399-404.

Higgins, D. (1966) Intermedia. Something Else Newsletter, 1, No. 1. Something Else Press, New York City. Reprinted with notes by Hannah Higgins in Leonardo, Vol. 34, No. 1, pp. 49–54 (2001). MIT Press.

Higgins, D. (1976) Some Poetry Intermedia, Unpublished Editions, New York City. Reprinted in the anthology Intermedia, Fluxus, and the Something Else Press: Selected Writings by Dick Higgins (2018), Siglio, Catskill, New York.

Higgins, D. (1987) A Short History of Pattern Poetry. In Higgins, D., *Pattern Poetry: Guide to an Unknown Literature.* State University of New York Press, Albany.

Ox, J. (2015) *Analogy and Conceptual Blending are Algorithmic Processes that Form a Toolkit for the Representation of Data.* RGS-IBG Annual International Conference 2015, University of Exeter, Devon, UK, September 2015.

Prusinkiewicz, Przemyslav and Artistide Lindenmayer, The Algorithmic Beauty of Plants. (1990) Springer-Verlag.

Sorensen, Vibeke and Rand Steiger, NLoops (1989). Algorithmic animation with musical composition derived in part from animation code. https://youtu.be/2EOMz0IZsGM?si=ooHEe_4zGlFtC4_8

Whitney. J. (1973) Hex Demo. A lecture at Cranbrook included on a laserdisc of his works issued by Pioneer in 1984.

Whitney, J. (1980) Digital Harmony: On the Complementarity of Music and Visual Art; McGraw-Hill, Peterborough, New Hampshire.

Youngblood, G. (1970) Expanded Cinema. Dutton, New York.

Zielinski, S. (2006) Deep Time of the Media: Toward an Archaeology of Hearing and Seeing by Technical Means. MIT Press, Cambridge, Massachusetts.