SEE THIS SOUND

AUDIOVISUOLOGY

VOLUME 1. COMPENDIUM
An Interdisciplinary Survey of Audiovisual Culture

VOLUME 2. ESSAYS
Histories and Theories of Audiovisual Media and Art

Edited by Dieter Daniels and Sandra Naumann
with Jan Thoben

Verlag der Buchhandlung Walther König, Köln

SEE THIS SOUND

Edited by
Dieter Daniels and Sandra Naumann
with Jan Thoben

Graphic Concept, Layout, and Typesetting
METAPHOR/Natalie Dietrich, Michael Jung,
www.metaphor.me

Translations (Compendium)
Gloria Custance (Introduction)
Jill Denton (Conceptual Correlations, Structural Analogies)
Aileen Derieg (Dance, Games; Color-Tone Analogies)
Rebecca van Dyck (Abstract Film, Architecture, Color Organs, Expanded Cinema, Film Scores, Gesamtkunstwerk, Graphic Notation, Sonification, Sound Design, Video; Parameter Mapping, Synchronization, Synesthesia)
Steven Lindberg (Literature, Music Video, Musical Theater; Audiovisual Perception)
Gillian Morris (Artists-Musicians; Transformation)
Niamh Warde (Cinedance, Interactive Art, Painting)
Hilla Widemann (Music, Sound Art)

Translations (Essays)
Gloria Custance (Daniels)
Jill Denton (Chion)
James Gussen (Diederichsen)
Steven Lindberg (Schneider essay)
Niamh Warde (Beller, Hóller, Introduction, Kwastek, Naumann, Schneider plates)

English Copyediting (Compendium)
Ginger A. Diekmann
Lutz Eitel
Niamh Warde

English Copyediting (Essays)
Ginger A. Diekmann

Editorial Coordination and
German Copyediting
textstern*/Ulrike Ritter
www.textstern.net

Proofreading
Lutz Eitel

Licensing and Copyright Management
Heike Helfert

Production
Printmanagement Plitt, Oberhausen

Lithography
Pixelstorm, Vienna

Published by
Verlag der Buchhandlung Walther König, Köln
Ehrenstraße 4, D-50672 Köln

Bibliographic information published by the Deutsche Nationalbibliothek. The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Printed in EU

Distribution
Germany & Europe
Buchhandlung Walther Kónig, Kóln
Tel. +49 (0) 221 / 20 59 6-53
Fax +49 (0) 221 / 20 59 6-60
verlag@buchhandlung-walther-koenig.de

Cornerhouse Publications
70 Oxford Street
GB-Manchester M1 5NH
Tel. +44 161 200 1503
Fax. +44 161 200 1504
publications@cornerhouse.org

Outside Europe
D.A.P. / Distributed Art Publishers, Inc.
155 6th Avenue, 2nd Floor
New York, NY 10013
Tel. +1 212 627 1999
Fax. +1 212 627 9484
eleshowitz@dapinc.com

ISBN 978-3-86335-613-2

An online version of this publication is available at www.see-this-sound.at

WEB ARCHIVE www.see-this-sound.at
developed and realized by the Ludwig Boltzmann Institute Media.Art.Research., Linz
Director: Dieter Daniels

Concept: Sandra Naumann
Editorial Team: Sandra Naumann, Jan Thoben
Technical Director: Mario Róhrle
Assistant: Bernhard Pusch
Design Web Frontend: Robert Pinzolits/
metaphor, www.metaphor.me
Database Programming and Web Frontend:
Matthias Tarasiewicz, Michal Wlodkowski/
5uper.net, www.5uper.net
Data Preparation and Entry: Adam Dörflinger, Carolina Hubelnig, Sandra Kobel, Linda Thalmann

Hosted at
Hochschule für Grafik und Buchkunst
Academy of Visual Arts
Leipzig

Audiovisual software art relies on computer software as its medium, and is primarily concerned with (or is articulated through) relationships between sound and image. Such works are produced for diverse social contexts and can serve a variety of objectives. In the field at large, and in the examples discussed in this article, software artworks serve some of the same aims as do cinema, performances, installations, interior design, games, toys, instruments, screensavers, diagnostic tools, research demonstrations, and even aids for psychedelic hallucination—though many projects blur these boundaries to such an extent that categorization may not be very productive. Likewise, audiovisual software artworks continue to emerge from plural and only occasionally intersecting communities of research scientists, new media artists, software developers, musicians, and isolated individuals working outside the institutions of the laboratory, school, museum, or corporation.

Owing to this diversity of origins and intents, the formal scope of what might be considered audiovisual software art is quite large as well. Some works generate images or animations from live or prerecorded sounds. Other projects generate music or sounds from static images or video signals, or use screen-based graphic interfaces to interactively control musical processes. Other artworks generate both sound and imagery from some non-audiovisual source of external information (e.g., stock trading data, human motion capture data) or from some internal random process. And yet other systems involve no sound at all, but are concerned instead with exploring the possibilities of visual music as an analogue to music in the visual domain. Many threads of influence and inspiration in the history of audiovisual software cut across these formal and technical distinctions. For this reason, this section considers audiovisual software art according to the principles of visualization and notation, transmutability, performativity, and generativity, which frequently motivate this kind of work.

Audiovisual Software Art

Golan Levin

Predecessors and Pioneers of Audiovisual Software Art

There are thousands or perhaps even tens of thousands of audiovisual software arts practitioners today, and yet the origin of these practices sprang from the work of just a handful of artists who obtained access to computer laboratories in the mid-1960s. The work of California-based filmmaker and animation pioneer John Whitney is a reasonable starting point for this history. Whereas most of Whitney's contemporaries in early computer art (such as Georg Nees, Frieder Nake, Manfred Mohr, and Chuck Csuri) were focused on software-generated plotter prints, Whitney was interested strictly in the music-like qualities of time-based imagery. Computers in the 1960s were too slow to generate complex images in real time, however, so Whitney instead used the computer to output frames of animation to film. In animations like *Permutations* (1966–1968; developed in collaboration with IBM researcher Jack Citron) and *Arabesque* (1975; created with the assistance of Larry Cuba), Whitney explored the ways in which the kinetic rhythms of moving dots could produce perceptual effects that were strongly analogous to modulations of musical tension. Whitney's films from this period were generally accompanied by nonelectronic music;[1] only later, with the advent of personal computing and real-time graphics in the early 1980s, did Whitney's focus shift to the development of a software instrument with which he could compose imagery and music simultaneously, as demonstrated in his animations *Spirals* (1987) and *MoonDrum* (1989).

Although the late 1960s and early 1970s witnessed rapidly accelerating technical advances in computer graphics and computer music, the capacity to generate *both* media in real time was still several years away. For this reason, many significant experiments that would lay the conceptual groundwork for purely computer-based real-time audiovisuals were nonetheless carried out in the offline context of the film studio. One example is the work produced at Bell Laboratories by the American computer artist Lillian Schwartz (b. 1928), who collaborated with notable computer musicians on abstract film animations like *MATHOMS* (1970, music: F. Richard Moore), *MUTATIONS* (1972, music: Jean-Claude Risset), and *MIS-TAKES* (1972, music: Max V. Mathews). Later, some artists combined computer-based synthesis in one medium with traditional live performance in another. German graphics pioneer Herbert W. Franke, writing in 1975, described the production of two ten-minute graphic music films (*Rotations* and *Projections*, 1974), in which live jazz musicians freely improvised in response to simultaneously projected patterns of abstract animated lines.[2] In Schwartz's 1976 performance, *ON-LINE*, a live dancer and musicians performed in real time while Schwartz, playing a QWERTY keyboard, created special graphic effects on a computer-controlled video system.[3]

Abstract Film, Animation

Live Visuals

1 John Whitney, *Digital Harmony: On the Complementarity of Music and Visual Art* (Peterborough: McGraw-Hill, 1980).

2 Herbert W. Franke, "Graphic Music," in *Artist and Computer*, ed. Ruth Leavitt (New York: Harmony Books, 1976), 83.

3 Lillian Schwartz; summary of works at http://www.lillian com/films/.

Possibly the first computer system capable of synthesizing both animation and sound in real time was the VAMPIRE (Video and Music Program for Interactive Real-time Exploration/Experimentation) system developed by Laurie Spiegel between 1974 and 1976 on a DDP-224 computer at Bell Laboratories in New Jersey. VAMPIRE offered a drawing tablet, foot pedal, and a large number of continuous knobs and pushbuttons—all of which could be used to modulate and perform a wide variety of image and/or sound parameters.[4] Built on top of Max Mathews's computer music research system, GROOVE (Generating Real-time Operations on Voltage-controlled Equipment), VAMPIRE, according to Spiegel, was an instrument for composing abstract patterns of change over time by recording human input into a computer via an array of devices the interpretation and use of each of which could be programmed and the data from which could be stored, replayed, reinterpreted and reused. The set of time functions created could be further altered by any transformation one wished to program, and then used to control any parameter of image or of sound (when transferred back to GROOVE's audio-interfaced computer by computer tape or disk). Unfortunately, due to the requirement of separate computers in separate rooms at the Labs, it was not physically possible to use a single set of recorded (and/or computed) time functions to control both image and sound simultaneously, though in principle this would have been possible.[5]

Parameter Mapping

Other significant software-based or software-generated audiovisual art of the late 1960s through the early 1980s are, among others, the computer animations of Stan VanDerBeek, Ken Knowlton, Tom DeFanti, and Larry Cuba; the computer-controlled laser projections of Paul Earls and Ron Pellegrino; and the interactive installations of Myron Krueger and Ed Tannenbaum. The introduction of the personal computer significantly broadened the landscape of audiovisual arts, making way for new forms like the digital video performance work of Don Ritter and Tom DeWitt, and the interactive desktop software creations of Adriano Abbado and Fred Collopy.

Interactive Art

Sound Visualization and Notation

Forty-three years after John Whitney's early experiments, real-time audiovisual software now comes as a standard component in every major computer operating system. At the time of this writing, a person's first encounter with audiovisual software is most likely to be with a screensaver (a software utility that prevents burn-in on some kinds of computer displays) or with a music visualization plug-in for a computer-based media player. In many cases, these functions are combined into a single piece of software. The aesthetics of such systems are more than occasionally targeted to a broad casual audience with an interest in psychedelic visual culture. The influential screensaver and visualizer *Cthugha*, for example, created by the Australian software developer Kevin "Zaph" Burfitt between 1993 and 1997, was advertized as "an oscilloscope on acid" and as a "form of visual entertainment, useful for parties, concerts, raves, and other events as well as just vegging out to mesmerizing, hypnotizing displays."[6] Despite this colorful language, *Cthugha's* self-description as an oscilloscope is actually quite accurate from a technical standpoint. An oscilloscope is a tool for viewing the waveform (or time-domain representation) of a signal, such as music, in real time—and *Cthugha* is essentially an elaborated oscilloscope which

---

4 Laurie Spiegel, "Graphical GROOVE: Memorial for the VAMPIRE, a Visual Music System," *Organised Sound 3* (1998): 187–191; also see http://retiary.org/ls/writings/vampire.html.

5 Spiegel, "Graphical GROOVE."

6 Kevin "Zaph" Burfitt, *Cthugha*, Winamp visualization plug-in, 1993–1997; http://www.afn.org/~cthugha/.

Video

decorates a sound's waveform by rendering it into richly colored variations of video feedback. Sound waveforms are the simplest possible information that can be extracted from digital audio data, and have therefore been used as the basis for numerous other visualizers as well, such as *Geiss* (1998–2008) and *MilkDrop* (2001–2007) by Ryan M. Geiss, *G-Force* (2000) by Andy O'Meara (which has been licensed for use in Apple's iTunes music player), *Advanced Visualization Studio* (2003) by Nullsoft, and *ProjectM* (2004) by Pete Sperl and Carmelo Piccione.[7]

Whereas some software artists have sought to produce entertaining or entrancing aesthetic experiences, others have approached the challenge of visualizing music to provide analytic insight into the structure of a musical signal. These works exchange the expressive visual languages of painting and abstract cinema for the conventions of legibility found in diagrams and music notation systems. An early example is Stephen Malinowski's *Music Animation Machine* (1982–2001), a software artwork which generated scrolling piano roll representations of MIDI sound files as a real-time graphic accompaniment to the music's playback.[8] The earliest versions of *Music Animation Machine* represented notes with colored bars whose vertical position corresponded to their pitch. Later variations of Malinowski's project incorporated additional visual schemata for representing the harmonic or dissonant qualities of musical chords, the spans of melodic intervals, and the timbres of different instrument tracks. Malinowski's system for showing note pitches is an example of a frequency-domain representation, which alongside the (time domain) waveform is the other principal mainstay of sound visualization systems. Frequency-domain representations take many forms, including piano rolls (so called because they resemble the paper scores used in nineteenth-century player pianos), spectrograms, sonograms, graphic equalizer displays, spectral waterfall displays, 3-D surface spectrograms, and (when applied to voice signals) voiceprints.

It is very common for audio visualization artworks, whether aesthetic or analytic, to present real-time animated graphics as an accompaniment to sound. Such systems typically display time-based representations of perceptual phenomena like pitch, loudness, and other relatively instantaneous auditory features. An interesting exception to this real-time trend is Martin Wattenberg's *The Shape of Song* (2001), a software artwork that produces static images from MIDI music in order to reveal its long-scale and multiscale temporal structures. For *The Shape of Song*, Wattenberg introduced an entirely new visualization method, termed "arc diagrams," which displays the ways in which constituent passages and phrases are repeated in a larger piece of music. *The Shape of Song* is necessarily a non-real-time visualization of music, as any real-time version would require perfect future knowledge of repetitions yet to happen.

The Transmutability of Data: Mapping Input Signals to Sounds and Images

Parameter Mapping

Sonification

A significant theme in many audiovisual software artworks is the transmutability of digital data, as expressed by the mapping of some input data stream into sound and graphics. For these works, the premise that *any* information can be algorithmically sonified or visualized is the starting point for a conceptual transformation and/or aesthetic experience. Such projects may or may not

---

7 "Music Visualization," on the website Wikipedia: The Free Encyclopedia, http://en.wikipedia.org/wiki/Music_visualization.

8 Stephen Malinowski, "Time-Line of the Music Animation Machine, 1970–2001," online at http://www.musanim.com/mam/mamhist.htm.

reveal the origin of their input data in an obvious way, and, indeed, the actual source of the transformed data may not even matter. This proposition is made particularly evident in *Data Diaries* (2002) by Cory Arcangel, in which the artist has used Apple's Quicktime movie player to straightforwardly interpret his computer's entire hard drive as if it were an ordinary movie file.[9] As Alex Galloway writes in the project's introductory notes, "[Arcangel's] discovery was this: take a huge data file—in this case his computer's memory file—and fool Quicktime into thinking it's a video file. Then press play."[10] Although Arcangel's process in *Data Diaries* posits a near total rejection of artistic craft, the results of his readymade technique nonetheless delineate a pure glitch aesthetic with a colorful and surprisingly musical quality.

Most commonly, the transmutability of data per se is not itself the primary subject of a work, but is rather used as a means to an end, in enabling some data stream of interest to be understood, experienced, or made perceptible in a new way. In such cases, the artist typically gives special attention to the aesthetics (and sometimes the legibility) of the audiovisually rendered information. The software artworks in the *Emergent City* series by the British artist Stanza are representative of this approach; these projects employ data collected from urban spaces as the basis for generating audiovisual experiences themed around cities. In *Datacity* (2004), a browser-based Shockwave application, sounds and video are collected in real time from multiple cameras around the city of Bristol, and are then collaged and manipulated to produce a "painterly interpretation of the landscape";[11] in *Sensity* (2004–2009), measurement signals from a network of wireless environmental sensors, deployed by the artist throughout his neighborhood, are used to generate audiovisual layers in an interactive map display.[12] The user of both projects is provided with various interfaces that allow further personalization of the audio mix and visual experience. Other artists have developed software art based on audiovisual mappings derived from weather data, network traffic (*Carnivore*, 2001, by Alex Galloway and the Radical Software Group),[13] seismic activity (*Mori*, 1999, by Ken Goldberg et al.),[14] ebay user data (*The Sound of ebay*, 2008, by Ubermorgen),[15] topographic data (*G-Player*, 2004, by Jens Brand),[16] and casualty statistics from the U.S. military action in Iraq (*Hard Data*, 2009, by R. Luke DuBois),[17] to name just a few examples.

The voyeuristic software installation *Listening Post* (2001), by Mark Hansen and Ben Rubin, produces a particularly moving audiovisual experience by mapping voice sounds and typographic images to text fragments culled "in real time from thousands of unrestricted Internet chat rooms, bulletin boards, and other public forums. The texts are read (or sung) by a voice synthesizer, and simultaneously displayed across a suspended grid of more than two hundred small electronic screens."[18] By rendering these otherwise disembodied texts into

---

9 Cory Arcangel, *Data Diaries*, 2002; http://www.turbulence.org/Works/arcangel/.

10 Arcangel, *Data Diaries*.

11 Stanza, *Datacity*, 2004; http://soundtoys.net/toys/datacity-2004.

12 Stanza, *Sensity*, 2004–2009; http://www.stanza.co.uk/sensity.

13 Radical Software Group (RSG), *Carnivore*, 2001; http://r-s-g.org/carnivore/.

14 Ken Goldberg, Randall Packer, Gregory Kuhn, and Wojciech Matusik, *Mori: An Internet-Based Earthwork*, 1999; http://www.ieor.berkeley.edu/~goldberg/art/mori/.

15 Ubermorgen, *The Sound of ebay*, 2008; http://www.sound-of-ebay.com/100.php.

16 Jens Brand, *G-Player*, 2004; http://g-turns.com.

17 R. Luke DuBois, *Hard Data*, interactive flash applet, 2009; http://transition.turbulence.org/Works/harddata.

18 Mark Hansen and Ben Rubin, *Listening Post*, software installation, 2001; http://www.earstudio.com/projects/listeningpost.html.

sounds and animated typography, this project literally *gives voice* to the unspoken words of thousands of people, placing its viewer at the center of a maelstrom of desires, opinions, chatter, and solicitations collected from around the world.

Animation

The design space of data-mapping projects has been humorously summarized in Jim Campbell's *Formula for Computer Art* (1996–2003), an animated cartoon diagram, which mischievously implies that the inputs to many data-mapping artworks may be fundamentally arbitrary and thus interchangeable.[19]

Mappings Based on Human Action: Instruments

A variety of performative software systems use participatory human action as a primary input stream for controlling or generating audiovisual experiences. These systems range from screen-based musical games, to deeply expressive audiovisual instruments, to puzzling and mysterious audiovisual toys whose rule-sets must be decoded gradually through interaction. In many cases the boundaries between these forms are quite blurry. Some of these systems are commercial products; others are museum installations or browser-based Internet experiences; and some projects have moved back and forth between these forms and contexts. What these applications all share is a means by which a feedback loop can be established between the system and its user(s), allowing users or visitors to collaborate with the system's author in exploring the possibility-space of an open work, and thereby to discover their own potential as actors.

Games

The category of performative audiovisual software games is extremely large, and is treated in depth elsewhere in this volume. Here I briefly note games that may also be intended or regarded as artworks, such as Masaya Matsuura's *Vib-Ribbon* (1999), a rhythm-matching game, or art/game mods such as *RC* (1999) by Retroyou (Joan Leandre), in which the code of a race-car game has been creatively corrupted and repurposed. One particularly notable gamelike system is *Music Insects* (1991–2004) by Toshio Iwai, which functions simultaneously as a paint program and a real-time musical composition system, and which Iwai has presented in a variety of formats, including museum installations and commercial game versions.

Numerous audiovisual instruments have been created which allow for the simultaneous performance of real-time imagery and sound. Many of these screen-based programs use the gestural, temporal act of drawing as a starting point for constructionist audiovisual experiences. A pioneering example of this was Iannis Xenakis's *UPIC* (1977–1994), which allowed users to gesturally draw, edit, and store spectrogram images using a graphics tablet; a 1988 version offered performance and improvisation of spectral events entirely in real time.[20] Whereas *UPIC* was developed to be a visually based instrument for composing and performing sound, other audiovisual performance systems have been explicitly framed as open works or meta-artworks—that is, artworks in their own right, which are only experienced properly when used interactively to produce sound and/or imagery. A good example is Scott Snibbe's *Motion Phone* (1991–1995), a software artwork that allows its user to interactively

19 Jim Campbell, *Formula for Computer Art*, 1996–2003; http://www.jimcampbell.tv/formula/index.html.

20 Golan Levin, "The Table Is the Score: An Augmented-Reality Interface for Real-Time, Tangible, Spectrographic Performance," in *Proceedings of the International Conference on Computer Music 2006* (ICMC'06), New Orleans, November 6–11, 2006.

create and perform visual music resembling the geometric abstract films of Oskar Fischinger or Norman McLaren. *Motion Phone* records its participant's cursor movements and uses these to animate a variety of simple shapes (such as circles, squares, and triangles), producing silent but expressive computer graphic animations.[21] A related artwork, Golan Levin's *Audiovisual Environment Suite*, or *AVES* (2000), presents a collection of cursor-based interactions by which a user can gesturally perform both dynamic animation and synthetic sound, simultaneously, in real time. Based on the metaphor of an "inexhaustible, infinitely variable, time-based, audiovisual 'substance' which can be gesturally created, deposited, manipulated, and deleted in a free-form, nondiagrammatic image space," Levin's system uses recordings of the user's mouse gestures to influence particle simulations, and then applies time-varying properties of these simulations to govern both visual animations and real-time audio synthesis algorithms.[22] Amit Pitaru's *Sonic Wire Sculptor* (2003) likewise produces both synthetic sound and animated graphics from the user's mouse gestures, but shifts the representational metaphor from a 2-D canvas to a 3-D space populated by the user's ribbonlike drawings.[23] Josh Nimoy's popular *BallDroppings* (2003) departs from free-form gestural interaction, presenting instead an elegant mouse-operated construction kit wherein "balls fall from the top of the screen and bounce off the lines you are drawing with the mouse. The balls make a percussive and melodic sound, whose pitch depends on how fast the ball is moving when it hits the line."[24] Nimoy articulately summarizes the hybrid nature of such work: "*BallDroppings* is an addicting and noisy play-toy. It can also be seen as an emergence game. Alternatively this software can be taken seriously as an audio-visual performance instrument."

Abstract Film

Interactive Art

Another genre of performative audiovisual software dispenses with drawing altogether, in favor of a screen space populated (usually a priori) with manipulable graphic objects. Users adjust the visual properties (such as size, position, or orientation) of these objects, which in turn behave like mixing faders for a collection of (often) prerecorded audio fragments. This can be seen in *Stretchable Music* (1998), an interactive system developed at the Massachusetts Institute of Technology by Pete Rice, in which each of a heterogeneous group of responsive graphical objects represents a track or layer in a precomposed looping MIDI sequence.[25] Other examples of this interaction principle can be seen in John Klima's interactive *Glasbead* artwork (2000), "a multi-user persistent collaborative musical interface which allows up to 20 online players to manipulate and exchange sound samples,"[26] or more recently in *Fijuu2* (2004–2006) by Julian Oliver and Steven Pickles, whose adjustable graphical objects allow for audio manipulations that are even more dramatic.

The systems described above were designed for use with the ubiquitous but limited interface devices of desktop computing: the computer mouse and the keyboard. The use of comparatively more expressive user interface devices, such as video cameras and custom-tangible objects, considerably expands the expressive scope of instrumental audiovisual software systems, but it also pulls them towards the formats (and physical dimensions) of performances and/or

21 Scott Snibbe, "The Motion Phone," in *Proceedings of Ars Electronica '96*, ed. Christine Schöpf; http://kultur.aec.at/lab/futureweb/english/prix/prix/1996/E96azl-motion.html.

22 Golan Levin, *Painterly Interfaces for Audiovisual Performance* (master's thesis, Massachusetts Institute of Technology, 2000); http://www.flong.com/texts/publications/thesis.

23 Amit Pitaru, *Sonic Wire Sculptor*, 2003; http://www.pitaru.com/sonicWireSculptor/.

24 Josh Nimoy, *BallDroppings*, interactive software, 2003; http://www.balldroppings.com.

25 Levin, *Painterly Interfaces*.

26 John Klima, *Glasbead*, interactive networked software, 2000; http://www.cityarts.com/glasbeadweb/glasbead htm.

installations. Finnish artist and researcher Erkki Kurenniemi's landmark 1971 *DIMI-O* system (Digital Music Instrument, Optical Input) synthesized music from a live video image by scanning the camera signal as if it were a piano roll.[27] David Rokeby's *Very Nervous System* (1986–1990), explored the use of camera-based full-body interactions for controlling the simultaneous generation of sound and image. Other audiovisual software instruments have employed custom-tangible objects as their principal interface, such as *Audiopad* (2003) by James Patten and *reacTable* (2003–2009) by Sergi Jordà, Marcos Alonso, Günter Geiger, and Martin Kaltenbrunner; both of these instruments use real-time data about the positions and orientations of special objects on a tabletop surface to generate music and visual projections.[28]

Generative Audiovisual Systems

The above sections have discussed several genres of audiovisual software artworks, including systems that use music to generate aesthetic or analytic visualizations, artworks that map real-world data signals to graphics and sound, and artworks that use human performances to govern the synthesis of animation and music. A fourth significant genre of audiovisual software artworks, known as "generative artworks," produces animations and/or sound autonomously—from their own intrinsic rule-sets. These rules may range from trivial forms of randomness to sophisticated algorithmic techniques that simulate complex organic processes or that even implement artificial intelligence models of visual and musical composition. One influential example of such an autonomous artwork is Scott Draves's *Bomb* (1993–1998), a free software system that produces fluid, textured, rhythmic, animated, and generally non-representational visual music.[29] *Bomb* uses recursive and nonlinear iterated systems, such as cellular automata algorithms (often used to simulate animal population behavior), reaction-diffusion equations (used to simulate organic pattern formations, such as leopard spots and zebra stripes), and video feedback. According to Draves, one of the most important innovations in *Bomb* was the idea "of having multiple CA [cellular automaton] rules interacting with each other," which allowed the program to generate and evolve a truly vast range of organic graphic configurations.[30]

Whereas *Bomb* is silent, Antoine Schmitt's *Nanoensembles* (2002) uses simple generative techniques to produce both sound and animation simultaneously.[31] Small animated visual elements move back and forth across the canvas in *Nanoensembles*, each at their own rate, each producing a simple looping sound whose volume is related to their speed and position. Because each element has its own unique cycle period, the motions of each eventually go out of phase—as do the sounds of each. The result is an ever-changing and effectively endless audiovisual composition.

---

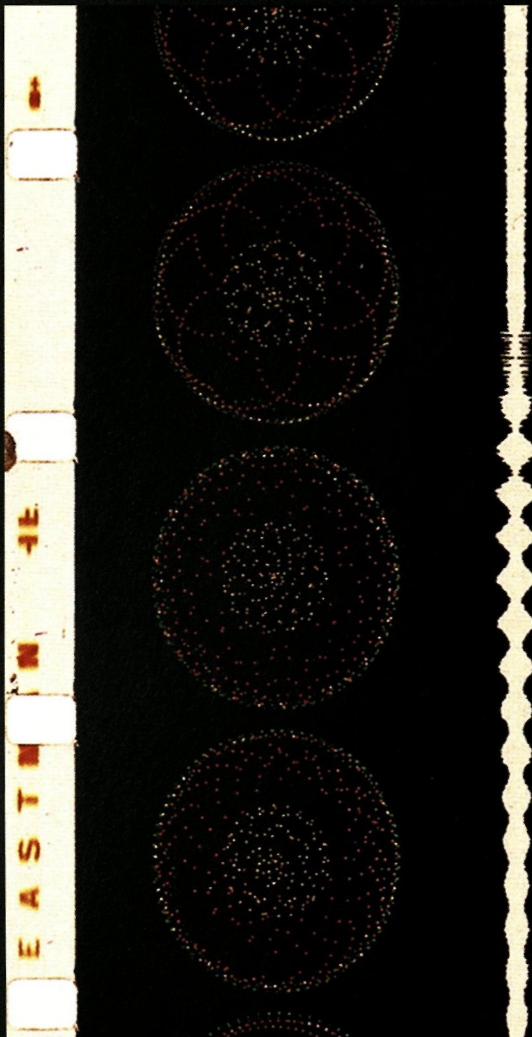
27 Levin, *The Table Is the Score.*

28 Tina Blaine, "New Music for the Masses," online at http://www.adobe.com/designcenter/thinktank/ttap_music/.

29 Scott Draves, *Bomb*, sound-responsive software, 1993–1998; http://draves org/bomb/.

30 Scott Draves, "Phylogeny of My Artwork and My First VJ Gig," January 27, 2006; http://draves.org/blog/archives/000361.html.

31 Antoine Schmitt, *Nanoensembles*, macromedia shockwave application, 2002; http://www.gratin.org/as/nanos/index.html, or http://soundtoys.net/toys/nanoensembles.



– Film strip from *Permutations* (1968) by John Whitney Sr.
© The Estate of John and James Whitney.

John Whitney
Permutations (1966–1968)

By the mid-1960s, John Whitney Sr. (1917–1995) had gained a significant reputation as a master of mathematically structured visual animation. It was in this context that, in 1966, the IBM Los Angeles Scientific Center granted Whitney its first artist in residence status in order to explore the expressive possibilities of the IBM Model 360 computer and IBM 2250 Graphic Display Console.

Whitney's work at IBM was supported by Dr. Jack Citron, a programmer and graphics researcher. In 1968, Whitney and Citron completed *Permutations*, an eight-minute computer graphic film wholly consisting of the independent circular movements of 281 colored dots.[1] For Whitney, the kinetic rhythms and phasing relationships of the dots' movements produced perceptual effects which were strongly analogous to modulations of tension in music:

> Every one of the points in *Permutations* is moving at a different rate and moving in an independent direction in accord with natural laws . . . Their action produces a phenomenon which is more or less equivalent to musical harmonics. When the dots reach certain numerical relationships with other parameters in the equation, they form elementary many-lobed figures. Then they go off into a non-simple numerical relationship and appear to be random again. I think of this as an order-disorder phenomenon that suggests the harmony-dissonance effect of music.[2]

Each frame of *Permutations* took approximately two seconds to compute, and was photographed to 16-mm film directly from the black-and-white IBM 2250 display. Color was added later, through the use of optical printing techniques. Upon its completion, *Permutations* was one of only a small handful of entirely computer-animated films in existence, and therefore influenced many subsequent animators interested in exploring the computer.

In his early computer works such as *Permutations* and *Arabesque* (1975), Whitney was above all focused on creating music-like structures in dynamic visual form—and not on creating mappings between image and sound. This focus changed after the publication of his book *Digital Harmony* in 1980: from then until his death in 1995, Whitney concentrated on the development of a software instrument on which he could compose visual and musical output simultaneously, in real time. In collaboration with programmer Jerry Reed, Whitney developed an audiovisual composing system, the Whitney-Reed RDTD (ca. 1980–1995), which allowed him to create "musical design intertwined with color design tone-for-tone, played against action-for-action."[3]

In the final decades of his career, Whitney distilled his model for understanding temporal structures to the notion of computational periodics, or harmonic agreements of periodic functions. In this regard he wrote:

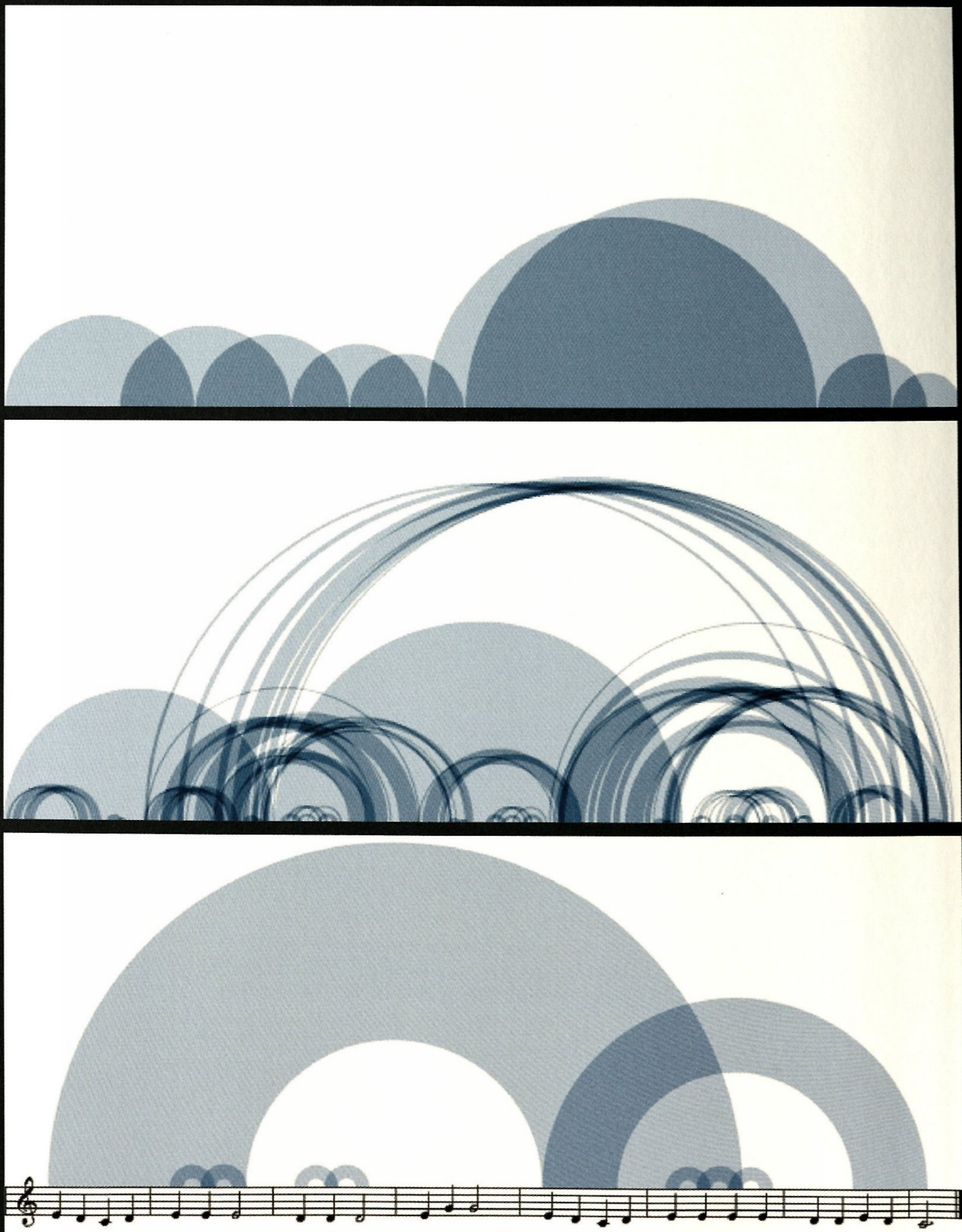
> Rhythm, meter, frequency, tonality and intensity are the periodic parameters of music. There is a similar group of parameters that set forth a picture domain as valid and fertile as the counterpoised domain of sound. This visual domain is defined by parameters which are also periodic. "Computational periodics" then is a new term which is needed to identify and distinguish this multidimensional art for eye and ear that resides exclusively within computer technology.[4]

1 John Whitney, *Digital Harmony: On the Complementarity of Music and Visual Art* (Peterborough, N.H.: McGraw-Hill, 1980), 196.

2 Ibid., 218.

3 John Whitney, "Fifty Years of Composing Computer Music and Graphics: How Time's New Solid-State Tractability Has Changed Audio-Visual Perspectives," *Leonardo* 24 (November 1991): 597–599.

4 John Whitney, cited in Ruth Leavitt, *Artist and Computer* (New York: Harmony Books, 1976), 80.



– Visualizations of Philip Glass's *Candyman 2*, Frédéric Chopin's *Mazurka in F# Minor*, and "Mary Had a Little Lamb" from *The Shape of Song* (2001) by Martin Wattenberg. To clarify the connection between the visualization and the song, in the diagram the score is displayed beneath the arches.

Martin Wattenberg
The Shape of Song (2001)

Martin Wattenberg (b. 1970) is an American artist and scientific researcher who creates visual treatments of "culturally significant data."[1] Best known for creating interactive data visualizations, Wattenberg is especially sensitive to the dialectic potential of information displays. His early works figured significantly in establishing the genre of artistic information visualizations, which in the meantime have become quite common. In 2001, with the help of a commission from Turbulence.org, Wattenberg turned his attention to visualizing the structure of music through his project *The Shape of Song*.[2] Many audiovisual software artists have attempted to visualize music with the help of real-time graphics; Wattenberg, on the other hand, has approached the problem as one of notation and produced a project that develops a new form of static image revealing hidden patterns latent in the music's score. Indeed, *The Shape of Song* is necessarily a non-real-time visualization of music, as any real-time version would require perfect future knowledge.

The Shape of Song is specifically designed to reveal repetitions of musical passages within MIDI files. The project takes the form of an online Java applet, within which visitors can select a preloaded musical composition (e.g., Bach, The Beatles, Britney Spears) or upload a MIDI file of their own for visualization. Wattenberg explains the straightforward method as follows:

> The diagrams in *The Shape of Song* display musical form as a sequence of translucent arches. Each arch connects two repeated, identical passages of a composition. By using repeated passages as signposts, the diagram illustrates the deep structure of the composition. For example, the picture above was built from the first line of a very simple piece, "Mary Had a Little Lamb." Each arch connects two identical passages. To clarify the connection between the visualization and the song, in this diagram the score is displayed beneath the arches . . . The resulting images reflect the full range of musical forms, from the deep structure of Bach to the crystalline beauty of Philip Glass.[3]

In a software landscape heavily populated by real-time displays of pitch, loudness, and other instantaneous auditory features, *The Shape of Song* continues to be a significant approach to music visualization because of its ability to represent long-scale and multiscale temporal structures in music. Since the project's launch in 2001, Wattenberg's Arc Diagram visualization method has been particularly influential in the broader field of information visualization as well, finding new uses in projects for visualizing such diverse datasets as Internet chatting behavior, hypertext links between blogs, biblical cross-references, and campaign contributions.[4]

---

1 Martin Wattenberg, personal website, http://www.bewitched.com/.

2 Martin Wattenberg, *The Shape of Song*, online interactive artwork; http://turbulence.org/Works/song/.

3 Wattenberg, personal website.

4 See http://www.visualcomplexity.com/vc/index.cfm?method=Arc%20Diagrams; and "Arc Diagrams: Visualizing Structure in Strings," *Proceedings of the IEEE Symposium on Information Visualization* (InfoVis'02), 2002, online at http://www.research.ibm.com/visual/papers/arc-diagrams.pdf.

– Screenshot and installation view *Music Insects* (1992) by Toshio Iwai.
© Exploratorium, http://www.exploratorium.edu. Photo: Amy Snyder.

Toshio Iwai
Music Insects (1991)

Toshio Iwai's *Music Insects* is singular for the seamless way in which it hybridizes a pixel-based paint program, a real-time system for composing and performing music, and a wholly visual programming environment for animated behaviors.[1] The core interaction logic of this inventive software application is a graphic step-sequencer wherein animated graphical bugs trigger musical notes and/or change their direction of movement when they encounter fat, colored pixels placed in their path by the user. These colored squares define the audiovisual terrain for the four virtual bugs that crawl across the grid surface of the canvas. When one of these bugs encounters a colored square, a diatonic musical note is triggered whose pitch is linked to the square's color. Each bug represents a different musical instrument and has its own timbre with which it sonifies the squares; thus, one bug produces piano sounds when it collides with the pixels, whereas the other bugs produce percussion, bass guitar, or trumpet sounds.

The user can add, modify, and delete pixels while the bugs are engaged in performing the score. Additional sophistication is possible through the use of certain specially colored pixels, which have the effect of rotating or reversing the bugs that touch them. With these special pixels, the user can cause the bugs to create looping rhythms, phasing polyrhythms, and complex passages which seem to never repeat at all. In this way, the colored squares define a musical score whose results may be generative, unpredictable, highly rhythmic, or some combination of these. And, at the same time, the user of *Music Insects* is also authoring an image.

Iwai's *Music Insects* comes remarkably close to offering a completely balanced solution for authoring image and sound simultaneously. Iwai overcomes many of the legibility problems often associated with symbolic and diagrammatic scores, for example, through the use of his animated bugs, which act as self-revealing and self-explanatory "playback heads" for the live sound. Because the system's score elements are elementary pixels as opposed to well-defined symbols, moreover, the fine granularity of Iwai's audiovisual substance is well suited to the creation of abstract or representational images, and the system's display screen may be read equally well as a painting or a score.

A playful approach to audiovisual interaction design and musical experience such as that in Music Insects is a hallmark of many Japanese software projects and can also be observed, for example, in software artworks like Haruo Ishii's *Hyper-scratch* (various versions, 1993–2003) and Masaki Fujihata's *Small Fish* (2000).

1 *Music Insects* was originally developed as a screen-based museum installation for the Exploratorium in San Francisco.