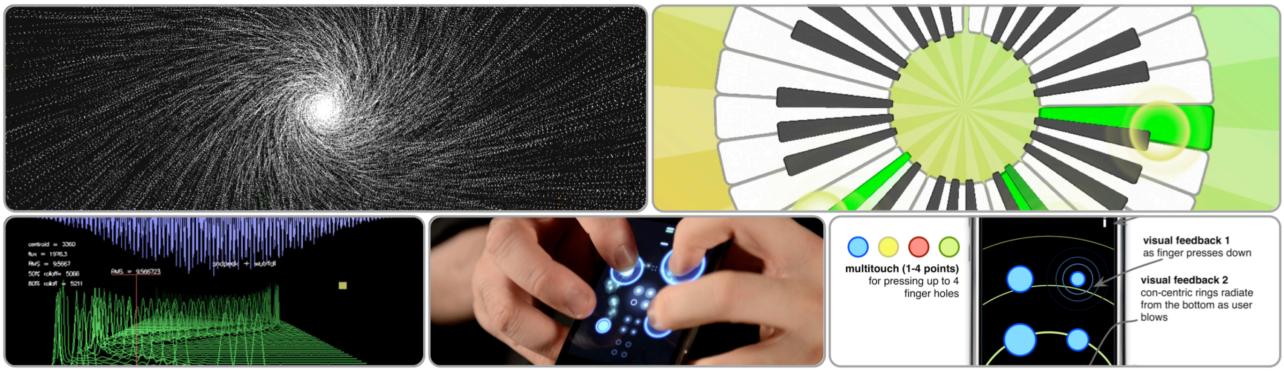
Principles of Visual Design for Computer Music

Ge Wang
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
`ge@ccrma.stanford.edu`



ABSTRACT

This paper presents observations pertaining to elements of expressive visual design for computer music, focusing in particular on real-time integration of graphics and audio. Specific projects are presented as examples and case studies supporting a set of design principles. They range from "user-oriented" to "aesthetic" and additional observations. Examples are categorized into audio visualization, game-like interfaces, and mobile instruments.

1. INTRODUCTION

We perceive and operate on multiple simultaneous modes of sensory input, including hearing (sound), sight (graphics), touch (interaction). These senses mutually reinforce each other and are essential in deriving expression, meaning, and aesthetic appreciation when creating and experiencing art. Of our senses, sight and hearing are most readily describable (and perhaps therefore most programmable on a computer). This paper focuses on this intersection of graphics and audio, and strategies for expressive visual design for computer music.

Through designing graphics-intensive computer music software systems, tools, and instruments over the last 10 years, the author has collected a set of principles for design and has developed a general philosophy. These principles are not intended to be universal (or necessarily original) but were arrived at through a sustained, iterative process of designing graphical computer music systems. These observations are targeted towards designers of computer music instruments, apps, and audiovisual software, and serve to provide some rules of thumb, and food for thought. Through a set of examples of specific software, instruments, and pieces, this paper aims to bring these principles to light. To set the tone, the principles are listed below – categorized into "user-oriented", "aesthetic", and "other". They will be referenced as appropriate from the examples.

Ge Wang, 2014. "Principles of Visual Design for Computer Muic." International Computer Music Conference.

<u>Some User-oriented Design Principles</u>

1) Real-time: make it so whenever possible
2) Design sound and graphics in tandem: neither should be an afterthought; seek salient mappings
3) Invite the eye – of experts and newcomers alike
4) Induce viewer to experience substance, not technology; hide the technology
5) Do not be afraid to introduce arbitrary constraints
6) Graphics can reinforce physical interaction (especially on touch screens)

<u>Some Aesthetic Principles</u>

7) Simplify: identify core elements, trim the rest
8) Animate, create smoothness, imply motion: it is not just about how things look, but how they <u>move</u>
9) Be whimsical, organic: glow, flow, pulsate, breathe: imbue visual elements with personality
10) Aesthetic: have one; never be satisfied with "functional"

<u>Some Additional Observations</u>

11) Iterate (there is no substitute for relentlessness)
12) Visualizing an algorithm can help to understand it more deeply (and can suggest new directions)
13) Video games, movies (and just about anything) can offer inspiration for visual design

2. VISUALIZING AUDIO PROCESSES

2.1 sndpeek and rt_lpc (2003-2004)

`sndpeek` began as a personal hacking project to make a simple teaching tool to visualize waveforms and Short-Time Fourier Transforms (STFTs) in real-time, using the microphone input (Figure 1). Accidentally, I noticed how the real-time visual response to sound encouraged small children (I was hacking at an extended family gathering) to "experiment" by vocalizing many sounds, eventually escalating into full-on screaming. Without any prompting from me, they intuited that higher frequencies appeared to the right on the STFT display (**principle #1: real-time whenever possible**, and **#3: invite the eye – of experts and newcomers**). There was even, briefly, a competition for who can scream higher (in both loudness and pitch), before the adults came in to put an end to the enterprise.

In `sndpeek`, the direct and immediate mapping between sound and graphics is apparent – the time-domain waveforms are simply drawn from each buffer of audio as it arrives from the microphone. A Fast Fourier Transform (FFT) is taken for each buffer, and the magnitude of each of FFT bin make up the spectral display. A sliding history of the STFTs is kept and animated in a scrolling waterfall plot [1]. The tool relied on the smooth animation and motion to convey information **(#8: animate, smoothness, imply motion**).

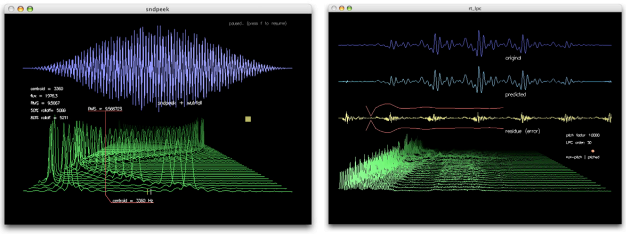


Figure 1. real-time audio visualization; (left) `sndpeek`'s waveform and waterfall plots; (right) `rt_lpc` visualizing various stages of LPC analysis and resynthesis.

We also implemented a successor to `sndpeek`, called `rt_lpc` (Figure 1), a real-time linear-predictive coding (LPC) visualizer that conveys stages in the LPC analysis, source/filter transformations, and resynthesis via periodic pulse train and all-pole filter derived from LPC coefficients. In creating this visualizer, the best takeaway for us was that we had gained a much more complete understanding of the algorithm **(#12: visualizing an algorithm helps to *grok* it – and suggests new directions**). We then added a real-time visualization of the vocal tract shape as sections from the LPC coefficient using Durbin Recursion. More often than not, designing visualization for a system seems to compel its designer(s) to really understand its process.

2.2 The Audicle (2004-2006)

The Audicle [2] was an ambitious attempt at deep integration between real-time visualization and the inner working of the ChucK audio programming language [3]. Implemented entirely in C++/OpenGL (as are most examples in this paper), the Audicle provided multiple views of core elements of the ChucK virtual machine (VM), including timing, processes (“shreds”), and scheduling (“shreduling”). Stats were tracked deep within the ChucK VM, and conveyed to the Audicle for visualization. Originally envisioned to facilitate live coding performances as a type of “program monitor as performance art”, the Audicle also contained an audio visualizer directly drawing from the real-time audio synthesis in ChucK's audio engine, and an animated, physics-based code editor **(#1: real-time whenever possible**).

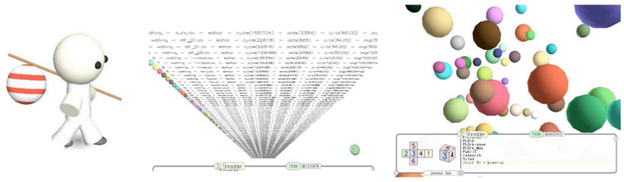


Figure 2. The Audicle visualizing various elements of the inner workings of a ChucK program; the center and right pane show active and recent processes.

Although the Audicle itself was ill-fated as an integrated programming environment, it spawned the simpler and much more successful miniAudicle [4], and served as the foundation for later laptop orchestra graphical interfaces. It was also a proof of concept for deep integration between graphics and complex audio environments in a real-time context – and the challenges therein.

2.3 Converge (2010-2012)

Created as part of an audiovisual composition for the Stanford Laptop Orchestra [4], the Converge visualizer was designed to be a “visual blender” of hundreds of images and associated location data, timestamp, and user descriptions – all collected using mobile phones from users in their daily life [5] (Figure 3). The piece was an exploration of the moments/sounds of daily life, memory, and passage of time. Each image had a live timer that, when enabled, highlighted our perpetual movement away from past moments, e.g.,”2 days, 5 hours, 28 minutes, 3 seconds ago… 4 seconds ago…”.

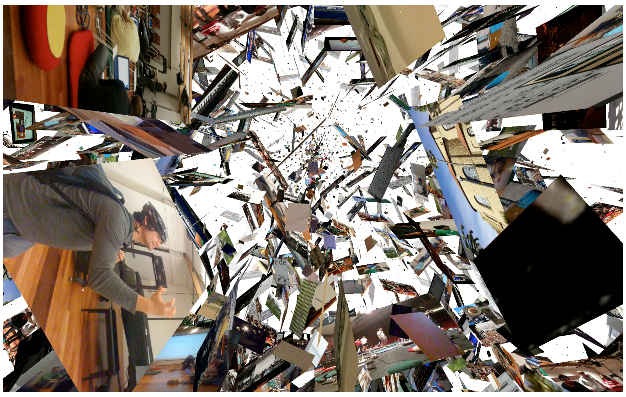


Figure 3. *Converge* visualizing and “blending” hundreds of user-generated photos.

An organic visual gesture involved exploding each image into 400 image fragments, all subject to a gravitational field that accelerated into a spiraling galaxy-like vortex (Figure 4 and **#9: don't be afraid to be whimsical and organic**). The shards can reform into their original images, or crumple into imploded balls of image fragments (playing on the imperfections and idiosyncrasies of memory). While these images were gathered prior to the performance, and so were not necessarily personal to the audience, they produced strong emotional response, possibly because the images were “mundane” moments of everyday life that all can relate to (**#4: hide the technology; induce viewer to think about substance**).

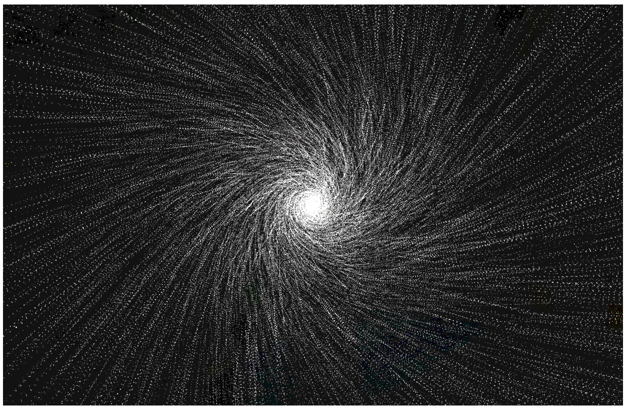


Figure 4. *Converge*'s "galactic vortex" visualization, comprised of thousands of image fragments.

3. GAME-LIKE INTERFACES

3.1 Non-specific Gamelan Taiko Fusion (2005)

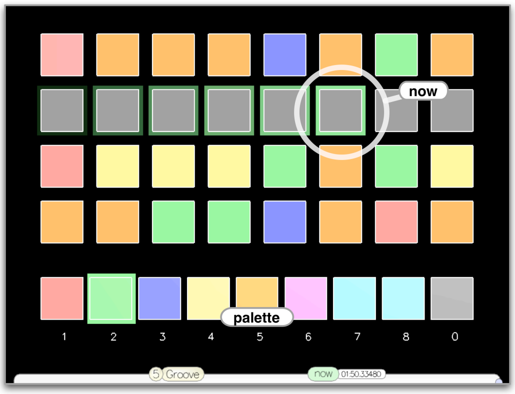


Figure 5. Interface for Non-specifc Gamelan.

Non-specific Gamelan Taiko Fusion by Perry Cook and Ge Wang was one of the very first pieces created laptop orchestra [6], and featured a local-area networked and synchronized step sequencer. The ensemble is divided into four sections; each of which can place any of eight timbres, represented by colored squares, on the 8x4 sequencer grid. An animated cursor washes over the sequencer, as a human conductor issues instructions to each of the sections regarding timbre and density. The interface was simple (**#7: identify core elements to visualize, trim the rest**). The animation, while minimal, was designed to make an intrinsically discrete step sequencer feel slightly more fluid.

3.2 ChucK ChucK Rocket (2006)

ChucK ChucK Rocket by Scott Smallwood and Ge Wang was a collaborative instrument inspired by the game *Chu Chu Rocket* [7] (**#13: video games can offer inspiration**). Players instantiate mouse-like critters onto a game board, directing them with arrows. As a critter runs over objects, sounds are made. Visually, the critters dart smoothly over the game-board; however, the underlying system is a discretely timed grid. Furthermore, all computers in the ensemble are networked synchronized, making complex interlocking rhythms possible. Graphically, an interpolator takes the synchronization signals and computes a velocity to animate the movement of each critter, giving the appearance of smoothness (**#7: animate, imply smoothness, motion**) –offering an essential game-like visual and interaction.

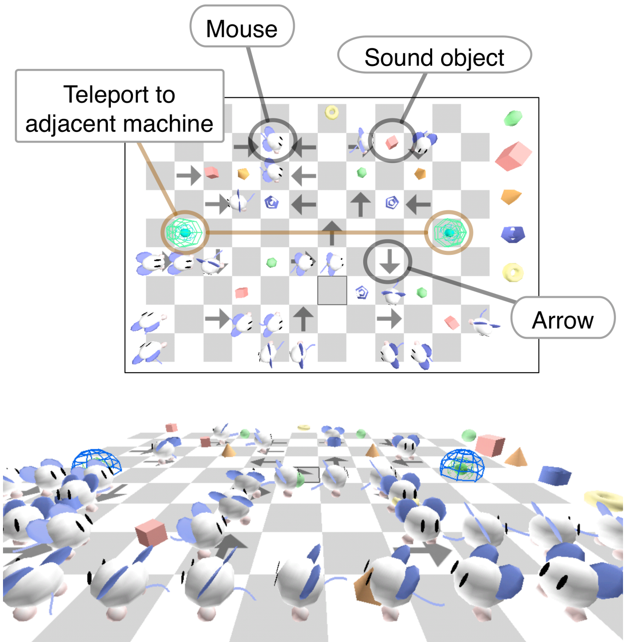


Figure 6. *ChucK ChucK Rocket* in action; (top) top-down view of game board; (bottom) perspective view.

3.3 LUSH (2010)

Principle #9 encourages experimentation with the **whimsical** and **organic**. *LUSH* [8] offers a fairly literal interpretation with an "ecosystem" where schools of musical entities (i.e., "fishes") roam according to group flocking algorithms. Each entity is encoded with musical information derived from non-deterministic finite automata, and makes its way through patches of musical triggers. A key aspect of *LUSH* is the aesthetics of motion and movement of the flocks gracefully roaming through the system, adding an organic element of chance.

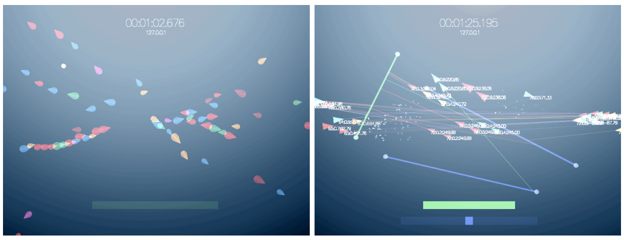


Figure 7. The LUSH musical eco-system.

4. INTERLUDE: INSPIRATIONS

In designing visuals for computer music systems, outside inspiration can suggest new functional, aesthetic, and technical directions. I believe that inspiration can and should be taken from wherever one may happen to find them. More obvious sources include synthesis or physical modeling parameters, aspects of the sound itself (e.g., waveform, spectral data, features). Less obvious sources of inspirations come from video games, cartoons, movies, or simply how a branch might sway in a breeze.

Gratefully, the author has drawn inspiration from such movies as Disney's *Fantasia* [9]. Consider, for example, "Sorcerer's Apprentice", where Mickey Mouse (the Apprentice) wields his absent master's magic wand in an attempt to "automate" his chores, only leading to chaos and mayhem (Figure 8). The visuals and animation design for this segment are meticulously and artfully tied to the musical score (by Paul Dukas 1896), a masterful example of conveying whimsy and magic through visuals and music (as brooms and other household objects take on personalities). At the same time, the Apprentice's role seems symbolic of our own as researchers and practitioners of a still nascent technology – it holds our fascination, and at times can feel like magic (and sometimes ends up in a mess).

Figure 8. Mickey Mouse conducting a symphony of magic and mayhem in *Fantasia* (1940).

Other sources of inspiration come from Edward Tufte's insights on information presentation [10], to Toshio Iwai's musical games [11], and audiovisual suites like Golan Levin's Painterly Interfaces [12]; the latter documented excellent examples of designing interactive sound and graphics as a single entity (**#2: design visual and sound in tandem**), and foregrounding substance over machinery. In yellowtail (Figure 9), Levin imposes the mechanic of (**#5: introduce arbitrary constraints**) of recording the gesture associated with drawing strokes, and extrapolating this information to organically animate each stroke while maintaining the essence of how the initial stroke was drawn (**#8: animate** and **#9: be organic**). (All are examples of **#13: anything can inspire**.)

Figure 9. Levin's *Yellowtail* animates brushstrokes using information from the drawing gesture itself.

5. VISUAL DESIGN FOR MOBILE MUSIC

Mobile and other touch screen-based instruments offer yet another dimension for real-time graphics, since the display is also the surface of interaction, presenting unique opportunities to couple visual design with physical interaction design.

5.1 Ocarina (2008) and Ocarina 2 (2012)

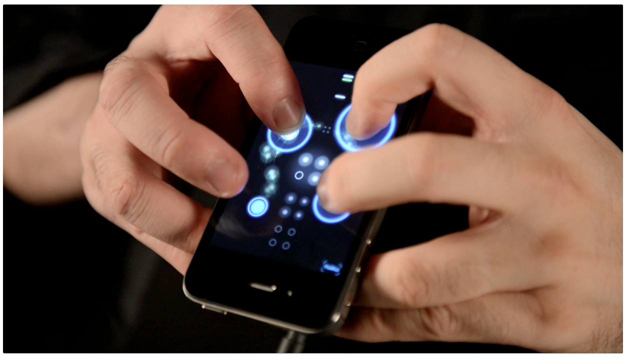


Figure 10. Animated fingerholes in *Ocarina* reinforce physical interaction by responding smoothly to touch, aimed to compensate for lack of tactile feedback.

Ocarina, designed in 2008 [13], was an exercise to create an expressive musical instrument specifically tailored to the iPhone (Figure 10). The physical interaction includes blowing into the microphone to articulate the sound, multi-touch to control pitch (via four onscreen virtual holes), and tilt to add and control vibrato. It embodies many of principles listed above, as it was designed to be visually inviting as an instrument/musical toy (**#3 invite the eye**). The visual design was also an exercise in reduction (**#7 simplify: identify core elements to visualize, trim the rest**), choosing to show only the functional fingerholes (and not the body of the instrument, which was also intended as a statement that the phone *is* the instrument) and visualization of breath and spinning particles in Ocarina 2 that respond to breath input (**#9 be whimsical and organic**).

Figure 11. *Ocarina*'s globe: listening to and visualizing other users around the world.

Much attention was devoted to the graphical interaction of the onscreen virtual fingerholes – they smoothly expand when a touch is detected. The goal was to make the experience feel responsive and also to compensate (to an extent) for the lack of tactility of the flat touch screen. Animated fingerholes help inform the user, often in their peripheral vision, that they have covered or activated a particular hole (**#6: graphics can reinforce physical interaction**).

The social aspect of Ocarina is presented through a visualization of the Earth displaying locations of recent Ocarina players around the world, which also highlights performance snippets (Figure 11). As accompaniment to the melody, a dual-helix animation emanates from the snippet's origin on the globe, and peacefully swirls into space, evoking both a sense of loneliness and connection. Functionally speaking, this visualization is perhaps completely unnecessary, but aesthetically it seemed completely essential to convey a sense of magic (**#10: have an aesthetic; never be satisfied with "functional"**), to hide the technology (**#4: induce viewer to think about substance, not technology**). The inspiration for the dual glowing helix actually came from a visual effect in video games, often used when a magic spell is cast or when a character gains a new ability (**#13: video games can offer inspiration**).

5.2 Magic Fiddle (2011)

The Magic Fiddle was designed specific for the iPad (Figure 12) and requires the user to hold the device near the chin and shoulder, like a violin [14]. The bowing interaction was replaced by an interaction that looks like a swirling vortex of smoke when touched, implying an active constant motion (**#5: introduce arbitrary constraints** and **#8: imply motion**). The graphics were, like Ocarina, designed to enhance a physical interaction (**principle #2: use visuals to reinforce physical interaction**) via responsive animations on the strings and the bowing region. We focused on only the core interactive elements (**#7: simplify**) and aesthetic elements (**#9: be whimsical and organic: glow**), including an additively blended neon glow on the strings and a flowing mist-like effect in the background, which also gave the visual effect of depth while emphasizing the virtual fiddle strings.

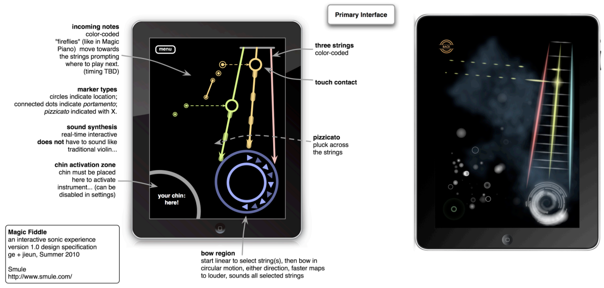


Figure 12. (left) Magic Fiddle design; (right) final version on iPad.

The visual and sound design for Magic Fiddle proceeded in tandem through many iterations (**#1: design sound + graphics in tandem** and **#11 iterate**), where the visual design stemmed from the parameters of the bowed string physical model (based on commuted synthesis), and the graphics guided the features of the sound synthesis, suggesting how glissandi and pizzicato might work. Ultimately, a glowing, neon-like aesthetic was adopted (**#10: have an aesthetic; never be satisfied with "functional"**).

5.3 Magic Piano (2010)

The core interaction in Magic Piano is exceedingly minimalistic (perhaps taking principle **#7: simplify** to an extreme), only involving falling flickering light particles (representing notes) and animated expanding rings in response to touch gestures (Figure 13). The lack of visible piano keys in this mode was in consideration of both the small touch screen size and the lack of tactility in distinguishing adjacent keys. The visual design, therefore, tried removing keys altogether, and encoded scores in the animated falling particles to be played expressively in time (**#5 introduce arbitrary constraints**).

Figure 13. Magic Piano's "songbook" mode; user controls timing to play pitches encoded in falling particles.

Interestingly, this "songbook" mode is by far the most popular in Magic Piano, one of Smule's most popular apps with more than 60 million users to date. By contrast, the "solo instrument" modes in Magic Piano (Figure 14) did feature visible piano keys, albeit contorted into various shapes, including a spiral that slowly "breathes", and a linear form that oscillates (**#9: be whimsical and organic: pulsate, breathe**). These whimsical modes were initial design experiments, and were left in as solo instruments. (Admittedly the contorted pianos are some of the most unnecessarily difficult-to-play instruments the author has ever designed – and experienced.)

Figure 14. The Magic Piano spiral keyboard mode: whimsical (and notoriously difficult to play).

5.4 Additional Examples and Resources

Those seeking additional, related audiovisual instruments can be found in *MadPad* [15] – a crowdsourced audio-visual sampling instrument, *Leaf Trombone: World Stage* [16] – a crowdsourced social music ecosystem, *Borderlands* [17] – a performance interface specifically designed to visualize granular synthesis (Figure 15), and more recently *Auraglyph* [18] – a handwriting driven graphics music programming environment.

Figure 15. Borderlands: a tightly coupled audiovisual granular synthesis laboratory.

6. CONCLUDING REMARKS

Visual design for computer music carries on in many different forms, yet perhaps much more as art than science. This is probably a good thing: it is difficult to envision a good design that blindly follows guidelines and without art or some spark of humanity. The principles here recur (some are much older than any works described here), and have bettered my work and the work of my friends and colleagues; yet they are not meant to stand alone – creativity and art are always essential. Perhaps these observations can serve as points of reference or departure as we collectively continue to explore the intersection of the sonic and the visual.

Acknowledgments

Thanks to Perry R. Cook for his mentoring on interaction and aesthetics, for his paper on the principles of computer music controller design [19] – this paper's structure is largely mirrored from it; to Philip Davidson, Spencer Salazar, and Mattias Ljungstrom for inspiration through countless audiovisual collaborations; to collaborators at CCRMA, Princeton, Smule, and elsewhere on works mentioned in this paper; to Jonathan Berger and Chris Chafe for encouraging me to write this paper; to Madeline Huberth for always insightful discussions of the aesthetics of computer music and helpful suggestions in crafting this paper.

7. REFERENCES

[1] Misra, A., Wang, G., and P. R. Cook. 2005. "SndTools: Real-time Audio DSP and 3D Visualization". *In Proceedings of the International Computer Music Conference*.

[2] Wang, G., P. R. Cook. 2004. "Audicle: A Context-sensitive, On-the-fly Audio Programming Environ/mentality". *In Proceedings of the International Computer Music Conference*. pp. 256-263.

[3] Wang, G. 2008. *The ChucK Audio Programming Language*. PhD Thesis, Princeton University.

[4] Wang, G., N. J. Bryan, J. Oh, and R. Hamilton. 2009. "Stanford Laptop Orchestra (SLOrk)." *In Proceedings of the International Computer Music Conference*.

[5] Oh, J. and G. Wang. 2011. "Converge: An Omni-Biographical Composition." *Computer Music Journal Emile*. Vol. 9.

[6] Smallwood, S., D. Trueman, P. R. Cook, and G. Wang. 2008. "Composing for Laptop Orchestra". *Computer Music Journal*. 32(1):9-25.

[7] *Chu Chu Rocket*. (Sega Dreamcast) Sega. 1999.

[8] Choi, H. and G. Wang. 2010. "LUSH: An Organic Eco-Musical System." *In Proceedings of the International Conference on New Interfaces for Musical Expression*.

[9] *Fantasia: The Sorcerer's Apprentice*. 1940. Dir. James Algar. Walt Disney.

[10] Tufte, E. 2001. *The Visual Display of Quantitative Information*. Graphics Press. Cheshire Connecticut.

[11] Iwai, T. 2004. "Images, Music, and Interactivity – the Trace of Media Art" *Keynote Speech, International Computer Conference on New Interfaces for Musical Expression*.

[12] Music Conference. June 2004. Levin, G. 2000. *Painterly Interfaces for Audiovisual Performance*. M.S. Thesis, MIT Media Laboratory.

[13] Wang, G. 2014. "Ocarina: Designing the iPhone's Magic Flute." *Computer Music Journal*. 38(2).

[14] Wang, G., J. Oh, and T. Lieber. 2011. "Designing for the iPad: Magic Fiddle." *In Proceedings of the International Conference on New Interfaces for Musical Expression*.

[15] Kruge, N. and G. Wang. 2011. "MadPad: A Crowdsourcing System for Audiovisual Sampling." *In Proceedings of the International Conference on New Interfaces for Musical Expression*.

[16] Wang, G., J. Oh, S. Salazar, and R. Hamilton. 2011. "World Stage: A Crowdsourcing Paradigm for Social / Mobile Music." *In Proceedings of the International Computer Music Conference*.

[17] Carlson, C. and G. Wang. 2012. "Borderlands: An Audiovisual Interface for Granular Synthesis." *In Proceedings of the International Conference on New Interfaces for Musical Expression*.

[18] Salazar, S. and G. Wang. 2014. "Auraglyph: Handwritten Computer Music Composition and Design." *In Proceedings of the International Conference on New Interfaces for Musical Expression*.

[19] Perry R. Cook. 2001. "Principles for Designing Computer Music Controllers." *ACM SIGCHI New Interfaces for Musical Expression (NIME) Workshop*.