The background is a complex, abstract composition of horizontal bands in various colors including green, blue, red, and yellow. Overlaid on these bands are several large, semi-transparent geometric shapes, primarily triangles and rectangles, in shades of green and blue. The overall effect is a layered, textured visual.

GLITCH -OMETRY

DANIEL TEMKIN
INTERVIEWED BY
CURT CLONINGER

ESSAY BY
DANIEL ROURKE

GLITCHOMETRY AN ESSAY BY DANIEL ROURKE

BY PLACING A GLASS PRISM in front of a shaft of sunlight and observing the resultant rainbow Isaac Newton was doing nothing new. Scholars of the Roman era, some millennium and a half earlier, had also known that such an intervention would produce such an effect.

Newton's original insight came from placing a second prism in the path of the rainbow, recombining the array of colours back into a single, white beam. His experiment disproved the common belief that the prism produced the colours from the sun's rays; that the purity of light was corrupted by the synthetic impurities of the glass.

Newton's experiment in 1666 signalled a science now known as spectrometry. By analysing the pattern and distribution of a spectra – whether of light or otherwise – information about the intervening environment can be derived. It is important to realise that it may very well have been impossible for Newton's revelations to come any earlier. For although the materials needed for his experiment had been around for millennia, the idea that a process of trial and error could be used to discern 'truths' about the world was a revolutionary conjecture. Whereas previously nature was considered to be at the constant mercy of God's will, in the 17th Century it was conceived that nature was hidden behind a cypher, and that experimentation consisted in making it appear.¹ As François Jacob succinctly put it, 'Any attempt to decipher nature and reveal its order requires certainty

that the grid will not change in the course of the operation.’² Not even God could adjust the laws of such a logical universe. Nature was a book, and human beings were such creatures driven to read it.

In Daniel Temkin’s *Glitchometry* series this tradition of experimentation endures. A ritualistic set of processes that consist in making discernible that which is hidden. Temkin’s wider practice takes place atop an innumerable series of cyphers, the distance between each often determined by technologies and technicities ordained by human accident, tradition or demand. Shunting the raw data of an image file into a program designed to manipulate audio – a process known as sonification – Temkin takes advantage of the potent potential of all digital ‘things’ to become something else. This mutability – transformations N. Katherine Hayles states ‘would be unthinkable if matter and energy, rather than informational patterns, formed the primary basis for... systemic exchanges’³ – allows different simulations to interfere, interweave and have their boundaries disturbed. *Glitchometry* blurs magic with experiment, alchemy with science, and the simulation with the thing itself. By treating the digital as if it were an originary nature Daniel Temkin may well be a deranged Isaac Newton and *Glitchometry* his alchemical decree.

Zeros and ones are themselves abstractions, cyphers designating the merest difference between one state and another. Raw digital information marks the silicon substrate of the computer as what Marcel Duchamp called the ‘infra-thin’, a ‘difference... reduced to the minimum needed to be technically perceptible’.⁴ And from these differences, as abundant as Moore’s law will allow, a perverse universe blooms. As Gilles Deleuze and Félix Guattari once exclaimed:

And when we consider what there is of a profoundly artificial nature... we cry out, ‘More perversion! More artifice!’ – to a point where the earth becomes so artificial that the movement of deterritorialization creates of necessity and by itself a new earth...⁵

As inhabitants of the ‘new earth’ we often encounter the infra-thin in the perceptual slip of a visual glitch, or the auditory pop-hiss that sometimes results from the compression of audio. Yet by their very definition glitches evade us, dissolving away at the instant of their unveiling. Built in to all digital architectures are a set of protocols that act like Maxwell’s Demon, categorising information as either pure signal or pure noise. This enforced binary, according to Mark Nunes, ‘imposes a kind of violence, one that demands a rationalisation of all singularities of expressions within a totalising system.’⁶

Out of this statutory logic – on/off; signal/noise – a digital nature is derived, but as glitch practices mischievously show, computation need not necessarily be reducible to the one or the other. *Glitchometry* turns away from the ‘new earth’; the milieu of cyphers that constitute our contemporary audio-visual cognizance. By foregoing the simulations relied on when Photoshopping an image Temkin assumes an almost meditative patience with the will of the digital. As with Duchamp’s infra-thin – ‘the warmth of a seat which has

just been left, reflection from a mirror or glass... velvet trousers, their whistling sound, is an infra-thin separation signalled⁷ – the one of the image and the other of the raw data is treated as a signal of meagre difference. Data is carefully bent in a sequence of sonifications that always risk falling back into the totalising violence of failure. The results

Data is carefully bent in a sequence of sonifications that always risk falling back into the totalising violence of failure.

sequence of words, distant from the contrivances they propose, over-signify the infra-thin, and trap the glitch in a glare that belies its every absence. The ritual then becomes one of composing new codes; of the juggling of metaphors and paradigms in the hope that at the very moment the balls collapse something ‘true’ might flicker into being. To grapple with *Glitchometry* one is invited to consider its failures productive, without transcoding them into success.

echo the aesthetics of glitch, in the jolt of pattern and the departures of colour, but as a practice *Glitchometry* shares more with the conceits of glitch: the desire to dance with something uniquely posthuman happening inside the machine, beyond even the limits of an infra-thin to capture.

Courting with *Glitchometry* is itself a paradoxical undertaking. This

DANIEL TEMKIN INTERVIEWED BY CURT CLONINGER

Q: I AM CURIOUS TO HEAR about your relationship with constraints, your “wrong” (ab)use of tools, and I’d love to hear your thoughts on the relationship between source code and surface visuals.

But before we get into all those things, I would like to know more about the nuts and bolts of the Glitchometry process. I know you are using an audio editor to modulate source image files, files that are seeded with geometric forms. If you don’t mind getting into some nerdity, which audio software, and how are you using it? Is it something like Ableton Live where you are kind of VJ-ing the source image and turning it into live video by tweaking knobs and sliders in real time and then you take a screen shot? Or are you doing a single tweak, then rendering the image and looking at it, and then re-tweaking the rendered version? What is your logistical process, and how tight is the feedback loop? As you work, how aware are you of the effect that your changes are having on the image?

A: *Glitchometry* images come from a long, iterative, tweaking and re-tweaking process, all within an audio editor (Cool Edit Pro – I’m one of thousands of Peter Quistgards). I’ve experimented with some of the other strategies you describe. At one point I had a fairly complex Max/MSP patch that would play each image as sound while I live-tweaked it, but I didn’t like the outputs in the end. The iterative process is slow and frustrating, but handing off control, back and forth with the machine, is the heart of the *Glitchometry* process.

I work with a variation of sonification, applying sound effects individually to each color channel. Much of my early glitch work, especially the Sector series, was made this way.

By using simple shapes (triangles, circles, etc), the textures in the photographs are no longer a factor, and all of the strange patterns that emerge are evidence of the process. It's intuitive: I have a vague idea of what each effect will do, but since I can't see the output, the process brings me somewhere unexpected. The longer I've worked with this project, the longer the feedback loop has become. I developed Glitchometry further during a residency at IEA (@ Alfred University). By this point, I had a pattern of working and feared having too much control of the image-making; as Sol LeWitt said, "When an artist learns his craft too well he makes slick art." I wanted to experiment with other techniques and also to make larger images with more detail. To avoid file-size maximums in Photoshop I began splitting huge images into one or two color channels to work on at a time, reconstructing them into redundant colorspaces; my favorite is seven channels: CMYKRGB. It meant spending weeks on an image instead of hours, and allowed me to work against that slickness and work larger.

What are your criteria for the decisions you are making? Avoiding overly large Photoshop file is one constraint; and you don't want the outcome to look "slick." But how are you determining the ways in which you tweak the software and the ways in which you massage the images? How do you know when an image is "finished"? Are you looking at the resultant images and determining that they are aesthetically pretty, or visually interesting, or generatively variable, and then deleting certain dead-end explorations that don't fulfill your criteria? I'm guessing you abandon certain moves in the audio software that yield less desirable results and pursue certain moves that tend to yield more desirable results. How are you determining what "desirable results" are?

First, I want to continually be surprised by the result. When I start a piece, I'm mostly breaking down the initial shape: adding noise, distorting its form, permeating it into the background. This topic phase doesn't last forever; if it did, the images would progress to a featureless grey static. Instead, new forms begin to crystalize, reacting to the initial shape and following the logic of algorithms designed for sound production. The wavy Moiré forms of flanger and dynamic delay (the ones I use for the *Glitchometry Stripes* pieces) are the most obvious of these. Another visualization of sound effects that arises regularly is the tripling of overlapped ellipses. These are what I try to bring out in the images as I work.

This entropic phase doesn't last forever: if it did, the images would continue to progress to a featureless grey static.

The software is a collaborator and tool. It has its own habits and an aesthetic, manifestations of algorithms designed for non-visual information. It also does not care what is recognizable or readable to us. My work is teasing out forms, experimenting with sound effects that clarify and build. So I work on an image until it no longer feels like it's moving forward, which usually means it's slipping away back

into noise or cycling through visually similar states. There's usually a point along the way when the image has the most visual clarity, and that's the one I pick as the final version. There are some that I go back to, like *Circles #5, #6, and #7*, which are all from the same sequence at different states.

I stop at a point that is surprising to me. I want to no longer be sure where I ended up or how. I don't want familiar abstractions or forms that suggest a subject. Some have seen nostalgia in *Triangles #3*, the red shapes like a topographical map, and the blue lines on yellow suggesting notebook paper we used as children. This is unintentional. The decision to freeze the image is intuitive and naive, based on the internal logic of the composition.

As you describe your process, it seems like you are searching for a kind of sweet spot of something (some "thing") between two extremes of nothing (no "thing") >> between: 1. an initial (silent) extreme of "nothing" prior to the image being massaged at all AND 2. an over-processed (noisy) extreme of "nothing" where the distinct forms disappear and head toward indistinctness (or banal repetition). So it seems like you are performing a kind of (Heideggerian) uncovering (unverborgenheit); more specifically, you are making visually manifest an otherwise invisible process.

Which leads me to two questions (involving code and photography and their relationship with each other): Glitchometry's process/outcome relationship seems very much related to a code/surface relationship, where the underlying code is analogous to the process, and the rendered visual surface is analogous to the outcome. Could you talk about Glitchometry from the perspective of a computer programmer? How is Glitchometry related to the other code-based art you create?

Yes, that description in terms of Heidegger is apt. There are two forms of irrationality I see coming together in *Glitchometry*. First, there's seeing meaning/narrative/representation in the noise, and then, against that, there's the algorithmic/systemic irrationality: the apparatus carrying out the same process forever, with no ability to regulate itself or to understand what's recognizable or meaningful as content. For me, this is an exploration of our relationship with logic systems, and with computers, which is the main way systematic thought manifests for us. I look at how artists like Gerhard Richter have dealt with similar conflicts. His *Abstraktes Bild* play off the viewers' hunger for recognizable images and patterns, which are part of the human process of looking. Yet the meaning of the pieces reside as much in showing how they're made and in refusing to give simple or understandable abstractions.

As a programmer, I'm used to commanding the machine at the code level. When something goes wrong, we perform a kind of detective work, guessing at why the software is behaving this way, and experimenting to isolate behaviors. Much of the work comes in understanding long chains of events set in motion with logical statements, with faith in a rational process underlying everything that happens, even if we can't find the cause. The process of misusing software with *Glitchometry* – allowing the machine to be in charge

for a while – undermines that rationality. Algorithms run wild. I see patterns but they are instinctual, vague. I try to allow the machine to function as an agent with its own agenda. I'm anthropomorphizing the machine, trusting it even as I see it as unhinged. This is freeing – I do my thing and the machine does its thing. I don't have to believe that figuring out why is even possible.

In a number of other projects, I've explored our relationship with machines, the compulsive way of thinking that evolves when we try to think logically. I see this compulsiveness as a learned behavior, our attempt to mimic a logical process, which is very far from how people actually make decisions.

My *Dither Studies* uses clashing colors to expose the ordinarily-hidden patterns of simple algorithms designed to hide compression. In *Straightened Trees*, I sort pixels in photos to make the trees perfectly symmetrical, warping the rest of the landscape. I've also worked with esolangs (esoteric programming languages) to address compulsiveness and irrationality. I wrote a language called Entropy, where all data decays as the program runs: each value alters slightly every time it is used, becoming less precise. In order to program effectively, an Entropy programmer needs to abandon the pursuit of precision that most programming demands. The output from an Entropy program is approximate, and the more the data is accessed, the more random it will become. At best, the programmer has a short window to get an idea across before the program corrodes.

The practice of uncovering and making visually manifest seems related to photographic darkroom techniques. Could you talk about Glitchometry from the perspective of a photographer? Are these Glitchometry images photographic? In what sense?

I'm fascinated in the difference between photographic and digital abstraction. As much as we try to make the second look like the first, there's a richness of light in Tillmans' *Abstract Pictures* or Jessica Eaton's photography in new-media drag that isn't quite there in digitally generated work. The first photography I really fell in love with is Stephen Shore's *Uncommon Places*, which is amazingly sensitive to this quality. I look to make work that falls in between.

I can see the darkroom analogy, but *Glitchometry* draws more from one of my first experiences in artmaking: My mother is a potter and, although I never gained proficiency with ceramics, I know well the highly ritualistic process of forming work. Each step feels counter-intuitive. What I love about the process is the familiarity that grows; the ritual guides progress in building something. You can make the same form over and over again and each time you get something unique. The glaze settles differently each time. Often, in digital practice, we use a tool and it's a wysiwyg experience, where the results are immediately visible. I think of ceramicists like Brother Thomas. He knew his kiln so well that he found certain spots in the kiln, which combined with glaze chemistry, produced colors that were impossible otherwise. He did so through practice, not through analysis of the process. With *Glitchometry*, I wanted to build something equally ritualistic: a set

of steps that I had to feel out at, leaving mystery in the outcome. I couldn't re-make a *Glitchometry* piece if I tried.

Some smart-alecky user interface designer (I forget who) said the way to make "interactive art" was just to look at all the decisions a good designer would make when creating a usable, intuitive interface; and then just make the exact opposite decisions. Presto! Interactive art! The declaration "inexact programming == software art" is obviously an oversimplification, but your approach does seem indebted to the 20th Century Dadaist project of absurdifying modernist rationality. As an artist, are you undermining our human rationality? Is that what glitch art is?

I hate to invoke (the much-abused) Incompleteness Theorems, but Godel (and Turing) have shown us definitively that "rationality" contains a mystery that's forever unsolvable. Logic is much weirder than it should be – when we can understand it, which is rare.

My work comes from my personal experience with the machine. I want the machine to be irrational the way that we are instead of the way that it is. I treat it like a stranger who doesn't speak the same language, but can understand broad gestures. There's a passage in Henry Miller's *Colossus of Maroussi* where he describes meeting the mayor of a mountain village in Greece. The two couldn't understand each other's language yet had a colloquy, entirely made up of imitations of stereotypes and stock characters. The next day, this mayor returned with a translator, allowing them to tell each other grandiose stories about their travels – all lies – until they eventually abandoned language to jump in the sea, where they "bit one another like crabs and screamed and bellowed in all the tongues of the earth."⁸

Through *Glitchomery*, I want this ecstatic meeting. For me, this is the most interesting aspect of James Bridle's *New Aesthetic* – the bizarre perspective we imagine when personifying the machine.

I'd like to speak to your question about glitch and *Glitchometry*'s relationship with glitch art. Clement Valla has said a number of times that his images "are not glitches. They are the absolute logical result of the system." This is interesting to me because a lot of glitch art is not really made up of glitches in a system at all, but rather unexpected input giving unexpected output through a system that works exactly as it should. Usually we're working with a process which, from the perspective of the machine, is functioning perfectly – it's just giving us results we're not expecting. It's we who are being glitched, not the machine.

You can see this in classic glitch techniques like databending/manual hex edits, etc (the ones *Glitchometry* is the most indebted to). For example, we can look at work like Ted Davis's *FFD8*, which exposes the code-behind-the-code for us to manipulate. For many of us, this was our first experience with glitch art, opening a JPEG in a hex editor. Here it's somewhat mediated in that Ted's program protects us from structural damage that might make the image undisplayable (the term 'FFD8' is named for the code which marks the

beginning of image data in the JPEG file format).

We have to remember here that JPEG is not a file format (the way the term is usually used), but an algorithm for compressing/decompressing visual data (the format is actually called JFIF). There's an algorithm at play in displaying the image. Even here where it's not one-to-one with pixel data because of the compression, our messing with data does not cause the algorithm to fail. We get an image that appears broken to us only because we expect the resulting image to make visual sense to us and it no longer does this. It's a similar effect in datamoshing; we can remove i-frames that establish new visual content, creating a flow of broken-seeming motion, the image suddenly disrupted. But the video "works," it plays perfectly fine. Again, it only appears broken to us because the input to that system is unusual.

If glitch art really broke things, this is what all glitch would look like:



The Glitchest Art

A lot of glitch art invokes a visceral sense of sabotage, whether it's "really" breaking something or not. *Glitchometry* goes in the other direction, exploring irrational but functional systems without seeing the result as digital failure – while using methods drawn from glitch practice (misuse of tools, manipulation of data, etc). It's an algorithmic art, it just purposely uses the wrong algorithm.

Your process for Glitchometry makes me think about Kandinsky trying to synesthetically paint music and winding up with purely abstract visuals. Here you are massaging visuals through audio software, and coming up with abstract visuals. Can you talk about the color of your pieces? I think of Kandinsky and the brilliant colors he used, a real contrast to Pollock's browns or Rothko's muted palette. Why are your colors so brilliant?

Also, can you talk about the resultant forms? Again, your forms seem to have more to do with

Kandinsky's abstract paintings (exploded forms, liquid forms), rather than the geometric forms of Rothko. To begin with basic geometric forms and then explode them seems like a kind of cubist approach. So why start with basic geometric forms? Is Glitchometry Cubist? Is Cubism proto-glitch? Modernist formalism sought to break down the complexity of nature into simple underlying forms. Glitchometry seems to be doing the opposite – beginning with source forms and then exploding them into much more complex outputs. Is it a kind of reverse Modernism?

I grew up admiring Op Art, the influence of which is apparent in work like my early *Dither Studies*. The *Glitchometry Stripe* pieces were influenced by Bridget Riley and Victor Vasarely. A series of differently (semi-randomly) curved lines in different colors has a dizzying optic quality, but it's enhanced by the way we most often see these images; scrolling them on the browser.

The Bauhaus artist I feel the most affinity for is Anni Albers. Her work has a strange ambivalence to the grid; the pieces which range from the maze-like forms made of right angles, to the later pieces that reference the grid more loosely, with more playfulness.

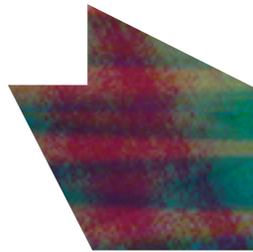
A few years ago, I made a Unicode hot-or-not project called *Unichar*. The idea was that selecting one Unicode character over another seems ridiculous, but once you make the first decision, you start to develop a system. I was shocked with the consistency of the results from different parts of the world; simple, asymmetric graphic shapes tended to do well, but so did anything that broke out of the grid. Unrepentantly rectangular forms did the worst, while figures that broke out of the bounds of their line of text did the best. For me, it was a big fuck-you to the artificial, modernist grid. In *Glitchometry*, in breaking down simple shapes, it feels like releasing them into an ambivalent space through this narrative of entropy – the liquifying of shape you referenced with Kandinsky.

How are these images different when printed versus as digital files versus as lightbox installations? Why print them? What happens to them when they are printed? Do they enter into dialogue with the history of printmaking? Do they somehow become less computery?

Lightboxes let me present something that's both not a screen and yet not exactly a print. It's monitor-like enough that the more pixelated forms feel at home, but it's not completely in the terrain of the computer. I wanted to present the work in a space between us and the machine.

Notes

1. Michel Serres, *The Birth of Physics* (London: Clinamen Press, 2000), 140.
2. François Jacob, *The Logic of Life: a History of Heredity; the Possible and the Actual* (Penguin, 1989), 30.
3. N. Katherine Hayles, *How We Became Posthuman : Virtual Bodies in Cybernetics, Literature, and Informatics* (Chicago Ill.: University of Chicago Press, 1999), 30.
4. Marcus Boon, *In Praise of Copying* (Cambridge, Mass.: Harvard University Press, 2010), 200
5. Gilles Deleuze and Félix Guattari, *Anti-Oedipus* (Continuum, 2004), 321.
6. Mark Nunes, *Error: Glitch, Noise, and Jam in New Media Cultures* (Continuum International Publishing Group, 2010), 5.
7. Marcel Duchamp and Paul Matisse, *Marcel Duchamp, notes* (G.K. Hall, 1983).
8. Henry Miller, *Colossus of Maroussi* (New Directions Books, 1941), 19.



Daniel Temkin, *Glitchometry*

On view at Transfer Gallery, Brooklyn, NY
November 16 – December 14, 2013

Interview by Curt Cloninger
lab404.com playdamage.org

Essay by Daniel Rourke
[@therourke](https://twitter.com/therourke) machinemachine.net