Final Draft of Final Paper
Samiha Amin
Research In The Disciplines 201
Professor Debra Keates
05/06/2016

Algorithmic Art and Music: Aiding Creative Expression

New technologies have enabled possibilities that were previously unthought of. In the field of computer art, code now replaces paintbrushes to create more than just still images. Algorithmic art, a subcategory of computer art, is implemented mainly through specialized programming languages such as Chuck and Processing, which exist primarily for the production of algorithmic art and music. This form of art relies on the computer executing code containing sets of operations called algorithms. While the artworks produced within the field of computer art are highly diverse and range from multimedia to robotics, this paper will focus on visual arts and music to examine the role of algorithms in the process of art-making.

For the purpose of a focused approach, this paper will consider algorithmic art as a method of computer art, directly incorporating algorithms in the process of creation, either by manually writing code or in a generative manner, using existing code. Live coding performances, where artists write code in real-time in front of an audience, incorporate algorithms that allow the live coder to conduct presentational musical performances. Moreover, software such as Polynomiography allows users to apply complex mathematical algorithms to generate visual artworks iteratively. In both cases, algorithms play a central role in the process of creation, while enabling a new medium for producing art. This new medium allows artists to experiment with and explore the boundaries of their creativity. However, the technology dependent nature of algorithmic art has its limitations, which can make algorithmic art intimidating on the surface.

This paper uses philosopher Immanuel Kant's definition of artistic genius in his *Critique of Judgment* as a framework for defining creativity. "Genius is the innate mental predisposition through which nature gives the rule to art" (Kant 174) shows genius is the natural ability to create original and exemplary work. This paper intends to explore whether algorithmic art and music aids individual creativity among artists or hinders it. This is done by examining the various ways creativity is impacted in the process of creation.

1. EXPANDING CREATIVITY

Generative art relies on iterative instructions contained in code. Ernest Edmonds, professor in the School of Software at the University of Technology Sydney, demonstrates the relationship between art and iteration in "What is Generative Art?":

> Making art is an iterative process. Art-making is a continual process of conception, action, and perception followed by reconceptualization--and so on. An important issue is that of matching the time and effort of the making (the laying down of a mark, for example) with its appropriate place in the cycle. Time-based work is very hard to make if each step has to be explicitly constructed. Using the computer as a generative engine magically removes this major problem. So, from a pragmatic point of view, I constructed a generative engine to make my time-based work so that the cycle of making, evaluating, and refining the work was faster and tighter. (6)

Edmonds separates the conceptualization process in art-making from the actions, which involve the time-based work. He argues that computers can facilitate art-making by carrying out the time-based work. This would allow the artist to focus on the conceptualization process, which leads to the visualization of the scenes involved and feeling of the emotional import of what is learnt from thoughts, leading to new forms of expression (Paschall 150). Therefore machines allow focus on creative thinking by carrying out the time-based work in the art-making cycle.

The idea of iterations in art-making Edmonds demonstrates is evident in Polynomiogra-

phy, which is an algorithm developed by Bahman Kalantari, a professor at Rutgers University in the department of Computer Science in “Polynomiography: A new intersection between mathematics and art”. The algorithm is “based on the use of one or an infinite number of iteration functions” (Kalantari 1) and takes in an input and applies the algorithm to generate visualizations, which the user can manipulate to create artworks. Edmonds would classify Polynomiography as a generative art form as it carries out the time-base work, which in this case are the calculations involving polynomials and the plotting of corresponding visualizations. Thor Magnusson, who is a researcher in computer music at the University of Brighton in UK, explains live coding in “Algorithms as Scores: Coding Live Music”. Performers start with a clean sheet and build their compositions from scratch. The performance involves writing of new code, changing, pausing or copying code blocks in order to transform it entirely. Simultaneously, visualizations are screened to compliment the code and music. Magnusson argues the code is in constant change and differs from traditional musical performances since “the code and the music evolve together in an interweaved process observed by the audience” (22). Therefore the computer generates the music while the coder provides the instructions in real-time. This means that the coder can skip learning and mastering traditional instruments and focus on the conceptualization process of music composition, which Edmonds explicitly separates from the iterative actions involved in art-making. In both Polynomiography and Live coding, artists are able to focus on the conceptualization process since the machine carries out the time-based work. Therefore algorithmic art aids individual creativity by allowing artists to focus on creative thinking in the art-making process.

Algorithmic art enables new possibilities to emerge by expanding individual creativity through the use of digital tools. Magnusson believes that the use of machines in music-making is

"analogous to the developments in painting when, with the advent of the photograph, the machine liberated painters from realism, resulting in impressionism and various other isms such as Cubism and Surrealism" (20). He is arguing that photography freed painters from realism, enabling them to utilize their creative thinking to produce impressionism, cubism and other unprecedented forms of art. This freedom pushed artists to create instead of just craft. This liberation through machines is also seen in Hession's case, which is discussed later, where machines allowed him to be experimental and work with sounds that were otherwise inaudible. Therefore machines allow new creative routes as a broader array of sounds as inputs are accessible to the artist, enabling an even greater range of possible outputs as products to explore. Thus the scope of the medium within which an artist could express is expanded. Jones, a doctoral researcher at Goldsmiths University of London, would agree with Magnusson as he found "the analytical, generative and adaptive features often found in these tools can offer new creative routes based on dynamic awareness of context and past history, harnessing the powerful probabilistic capabilities" of machines (176). Dynamic awareness influences greater-self-reflection, which Jones argues happens through a feedback loop described as the bouncing "back and forth between creative acts and reflection upon those acts" (183). The probabilistic capabilities refers to the ability of the computer to generate a broad range of outputs based on randomized inputs. Kalantari would agree on probabilistic capabilities of machines, since he argues his Polynomiography software enables users to produce "an infinite variety of designs … employed by an infinite class of complex polynomials" (2). Therefore machines provide a larger ground for the feedback loop to work within, further promoting creative thinking. Without machine liberation, the range of inputs and outputs in the art-making process would be relatively limited. Magnusson also empha-

sizes the role of tools leading to self-reflection, allowing inspiration and education, while describing live algorithmic art as a form that plays with the core of the tools used for artistic creation (22). Basing on Jones' idea "we live through our tools, and our tools shape our experiences" (177), which emphasizes that the tools used in art projects determine our experience of the art-making process itself, interacting with the core of the tools would provide a more in-depth understanding of the process and therefore even greater self-reflection. Since inspiration and education stems from self-reflection made possible through tools via feedback loops, tools are essential in the art-making process. Therefore algorithmic art aids natural creativity by allowing greater self-reflection and broadening the medium artists work within.

2. NON-LINEAR EXPRESSION

Code allows for self-expression, not just in the end result but also the through code itself. Geoff Cox, who is a professor in the school of Computing at University of Plymouth, and researcher and live coder Alex McLean look at code philosophically in "The Aesthetics of Generative Code". Their idea "code and the resultant actions are intricately linked in poetic dialogue" (Cox, McLean and Ward 12) suggests that code and the art it produces should be seen as a whole in the same manner in which a poem is seen. John Hospers, who was an instructor at Columbia University in Mathematics and philosophy, would agree with Cox as he describes art in "The Concept of Artistic Expression" as both the product and the process of creation (321). Since artists are able to express themselves through the code they write, they can explore and experiment with a broad range of tools and mediums they can choose from. In the case of algorithmic art, this becomes especially emphasized as a multitude of programming languages and tools exist already and are constantly advancing with developments in technology. This is seen significantly

in live coding where the code is projected on a screen for the audience to observe. Therefore the scope within which an artist could express creatively exceeds beyond just the end-product.

Code works like poetry in that it plays with the structures of language itself, as well as our corresponding perceptions and is generative in the sense that it unfolds in real-time (Cox, McLean and Ward 7). The generative property of poetry lies in the way poetry is structured in narration, which is similar to how code executes sequentially. Traditional visual art forms such as paintings and sculptures lack the narrative real-time flow that Cox associates with generative-ness, meaning computer art is rather an exclusive form which enables a flow of self-expression comparable to that of poetry. In both poetry and code, the understanding of the overall piece becomes clearer with every new line read or executed (by the computer in case of code). Hospers defines expression as an activity of the artist in the process of creation that is an exploration of the artist's own emotions as he argues "the artist has, as it were, deposited his emotions in the work of art, where we can withdraw it at any time we choose" (334)**.** Therefore, the flow of self-expression in poetry and code can be thought of as a stream of emotions that fluctuates in relation to time, similar to film and video art. Following Hospers idea of withdrawing emotions from art, the corresponding perceptions among the audience changes simultaneously as the frame of the poetry and code changes with each line. Similarly, the emotions withdrawn in the audience of generative art are in constant change as well, allowing a non-linear flow of expression which "presents elements of surprise" (Magnusson 21), instead of a static one that visual art forms generally allow.

One example of non-linear expression is seen in the *Electric Sheep Documentary*, which details software engineer Scott Draves' project. The project is a collective intelligence consisting

of 450,000 computers and people that use mathematics and genetic algorithms to create an infinite abstract animation. The animations are evolving gradually which generate from Draves' algorithm behind the software. The input-dependent characteristic of algorithm itself makes projects of such kind and scale possible. It can be argued that Draves' creativity reflects on every animated frame of the program even though he only coded the algorithm, not knowing all the possible animations his algorithm could render. This non-linear flow of expression enables the creativity of an artist to be applied and exhibited beyond just the end-product, especially in cases of evolving art as the end-product is in constant change.

3. TOOLS AND LIMITATIONS

Expression of creativity needs a medium to work within. McCormack argues "Individual creativity is arguably weak in the absence of the structures and systems that enable the accumulation of artifacts and information" (136). By using an example in terms of painting, this concept can be illustrated as the canvas acting as the medium, the artifacts representing the paint and the picture painted displaying the information. For computer art, the computers act as the structures and systems that human artists can utilize to strengthen their expression of individual creativity. However, artists need an understanding of the medium. Hospers cites John Dewey, author of *Art as Experience*, arguing that expression is "the artist's attempt to bend the medium to his will" (316). In order to bend the medium to one's will, the artist needs to have complete control over the medium. Without this grasp, artists will be unable to fully express themselves through their art.

In algorithmic art, computers are the medium whereas code is the tool. However, algorithmic art might not be easy to pick up, as researchers in the Computer Science department of

University of Otaga found “Programming courses are generally regarded as difficult, and often have the highest dropout rates” (Robins, J. Rountree and N. Rountree137). Programming is far from intuitive to use, unlike art tools such as paintbrushes and therefore falls behind in accessibility as “programming ability must rest on a foundation of knowledge about computers, a programming language or languages, programming tools and resources, and ideally theory and formal methods” (Robins, J. Rountree and N. Rountree 140). The extent of an artist’s programing abilities therefore can present as a limitation to his self-expression as it determines their control over the medium. Hospers would argue this can also complicate the audience perception, as the lack of control over medium can reduce the clarity of the intended artistic expression.

Furthermore, Cox argues “the world of multimedia is all too easily conflated with a multi-sensory experience” (3) where artists are concerned with making use of multiple mediums and combining still image, sound, touch, etc. With a large number of tools and mediums available to work with, artists may be tempted to use various technologies in their artwork, perhaps in excess of what is necessary. In order to do so, artists would have to learn to use each one of them and since learning of tools is an action-oriented task, Edmonds would argue that the attempt to incorporate various technologies solely to produce a certain result would shift the focus towards time-based work. Although algorithmic art can hinder individual creativity when artists try to make use of multiple technologies by limiting the focus on creative thinking, the extent of incorporation of technologies in artworks can be simplified by using them as tools for enhancement instead of the primary canvas.

4. ROLE IN THE PROCESS OF CREATION

While live coding is a stand-alone system and presents itself as an alternative to traditional music making, algorithmic art can also be used to enhance traditional practices. Paul Hession, a drummer working in jazz and free improvisation in the school of Music at University of Leeds, studied two cases to explore the role of live algorithms as an addition to traditional music-making in the article "Extending Instruments With Live Algorithms". First, he used analogue and digital tools to enhance his drum kit as a solo performer and ran algorithmic programs that sampled his input, randomized and replayed it. Using algorithms, he manipulated and accentuated tonal qualities of the drum set which would otherwise be inaudible. The algorithms also enriched the music by adding uncertainty. Second, he collaborated with researcher and live coder Alex McLean at live algorithmic performances, where they used the volume pedal to enhance the performance by muting signals. The muting allowed for concentration on the acoustic playing and production of an in-and-out fading effect at will, providing elements of surprise and more energy to the show. The research concluded that live algorithms enables new possibilities in traditional music-making by extending instruments in both solo and duo performances. Therefore, algorithmic art enables new possibilities that wouldn't be otherwise possible, as both a standalone system and as an addition to traditional music-making. Magnusson would argue this is a form of machine liberation, thus the extending of instruments aid individual creativity by allowing new creative routes.

The limitations of algorithmic art depends on the role it plays in the process of art-making. In his research, Hession found that live algorithms work as a "surrogate co-performer" (3) explaining their purpose as "to stimulate interaction, thereby simulating the push-and-pull that is intrinsic to playing with another human musician" (2). Therefore algorithms enable a mechanism

Fig. 1. Joshua Davis's Generative Art

similar to human collaborations, implying that an artist is able to apply techniques using live algorithms that would be only otherwise possible when playing with another human. The idea of a co-performer means it can be used as an addition instead of a replacement, discarding the scenarios where algorithmic art is limiting. This would work similar to when a human collaborator steps provides a helping hand in a certain project, adding to the project instead of replacing anything contributed by the original artist. Joshua Davis, an american designer and artist in new media, uses algorithms as an addition to his artwork, escaping the limitations that algorithmic arts may have. Seen in Davis's artwork (fig. 1), he combines hand-drawn shapes and algorithms in the process of creation. Therefore if an artist is faced with a technology they are unfamiliar with, pursuing it becomes a choice rather than a necessity. If the role of algorithmic art is to compliment an artist's already existing process, it does not impose the limitations that the artist faces when using technology as the canvas. Instead, algorithmic art allows to extend traditional mediums such as hand-drawing by enabling complex enhancements.

5. CONCLUSION

Algorithmic art and music present a field that is yet to claim its place in society, as currently it is only explored by academics, computer enthusiasts and new media artists. Algorithmic

art and music has potential that are often overshadowed by its apparent complexity. But it can prove to be extremely powerful and rewarding in processes of creations once its implementation is learned and the initial intimidation is overcome. Algorithmic art and music aids natural creativity by enabling focus on creative thinking in the art-making process, increased self-reflectivity due to the feedback loop and broadening of the medium. This form of art and music making also expands the scope within with an artist can express creatively, through enabling non-linear expression and expression through the code itself. While there are limitations to this art form since artists need to be proficient in the programming languages and tools they wish to use, these limitations can be dodged depending on how algorithmic art and music is implemented in one's process of creation, as it can be used both as the primary medium and as a tool for enhancement.

Thus, algorithmic art and music aids individual creativity in artists, by both increasing the innate human creativity itself and also the scope within which creativity can be applied. I learned from my research that Algorithmic art is often seen as unaccessible due to its technology dependent nature, but in reality it can very easily be implemented as a tool for enhancement without disrupting an artist's original workflow. While programming languages can be difficult to pick up, artists can use softwares like Polynomiography to incorporate algorithms in their artwork, which are fairly easy to use. Specific programming languages like Processing also exist that are designed to be an introduction to programming for non-coders. Therefore in the world of algorithmic art, tools exist to make it accessible to all skill levels. This would mean the process of art and music making would become more rewarding for artists, whilst multi-faceted, and artists would be able to express themselves more fully.

Algorithmic art acts as a bridge between the fields of art, mathematics and computer science. This allows the relationship between these fields to be seen more explicitly, as may not necessarily be obvious. Algorithmic art's visual nature and presentational performances involving algorithmic music, as seen in Live coding, can make computer science and mathematics more appealing, altering the heavily technical image generally associated with coding into a more creative and artistic one. Therefore the use of algorithmic art in education can potentially help to bring more people into STEM. Also, algorithmic art can turn the learning of these subjects into a more creative process, allowing STEM learners to explore their creativity simultaneously and facilitating learning of the subjects themselves. Overall, algorithmic art and music is a field worth exploring and over time, its implementation in artworks will hopefully be seen more frequently as more artists realize its potential, allowing for exciting new possibilities and creative discoveries.

Works Cited

Boden, Margaret and Ernest A. Edmonds. “What is generative art?” *Digital Creativity* volume 20 (1 & 2) March 2009: 21–46. Print.

Cox, Geoff, Alex McLean and Adrian Ward. “The Aesthetics of Generative Code.” In *International Conference on Generative Art* (2000). Web. 24 April 2016.

“Electric Sheep Documentary.” *Youtube.* Web. 24 April 2016.

Hession, Paul and Alex McLean. “Extending Instruments With Live Algorithms In A Percussion/ Code Duo”. *In Proceedings of the 50th Anniversary Convention of the AISB: Live Algo rithms (2014).*Web. 24 April 2016.

Hospers, John. “The Concept of Artistic Expression.” *Proceedings of the Aristotelian Society*, New Series, Vol. 55 (1954 - 1955): 313- 344. Print.

Jones, Daniel, Andrew R. Brown, and Mark d’Inverno .“The Extended Composer: Creative re flection and extension with generative tools.” *Computers and Creativity.* Ed. Jon Mc Cormack and Mark d’Inverno. 2012: 175–203. London: Springer. Print.

Davis, Joshua. Nasca Print / Edition 5, Anderson Ranch Art Complex, Snowmass Colorado. 2011. Image.

Kalantari, Bahman. “Polynomiography: A new intersection between mathematics and art” *Tech nical Report* DCS-TR 506, Department of Computer Science, Rutgers University, New Brunswick, New Jersey. 2002. Web. 24 April 2016.

Kant, Immanuel. *The Critique of Judgment.* Start Publishing LLC, 2013. Print.

Magnusson, Thor, "Algorithms as Scores: Coding Live Music." Leonardo music journal 21 (2011): 19-23. MIT Press. Print.

McCormack, Jon, et al.“Ten Questions Regarding Generative Art.” Leonardo, Vol. 47, No. 2. 2014:.135-141, MIT Press. Print.

Paschall, Alma. “What Is Creative Expression?” *The Elementary English Review* 11(6) (1934): 149–163. Print.

Robins, Anthony, Janet Rountree and Nathan Rountree. “Learning and Teaching Programming: A Review and Discussion” *Computer Science Education* Vol 13, No. 2. 2003: 137-172. Print.