The final authenticated version is available online at https://doi.org/10.1007/978-3-030-43859-3_4

Quantum Zentanglement: Combining Picbreeder and Wave Function Collapse to Create Zentangles®

Anna Krolikowski, Sarah Friday, Alice Quintanilla, and Jacob Schrum[0000-0002-7315-0515]

Southwestern University, Georgetown TX 78626, USA
{krolikoa,fridays,quintana,schrum2}@southwestern.edu

Abstract. This paper demonstrates a computational approach to generating art reminiscent of Zentangles by combining Picbreeder with Wave Function Collapse (WFC). Picbreeder interactively evolves images based on user preferences, and selected image tiles are sent to WFC. WFC generates patterns by filling a grid with various rotations of the tile images, placed according to simple constraints. Then other images from Picbreeder act as templates for combining patterns into a final Zentangle image. Although traditional Zentangles are black and white, the system also produces color Zentangles. Automatic evolution experiments using fitness functions instead of user selection were also conducted. Although certain fitness functions occasionally produce degenerate images, many automatically generated Zentangles are aesthetically pleasing and consist of naturalistic patterns. Interactively generated Zentangles are pleasing because they are tailored to the preferences of the user creating them.

Keywords: Zentangle · Compositional Pattern Producing Networks · Wave Function Collapse · Neuroevolution · Interactive Evolution

1 Introduction

Artificial intelligence can create unique and visually compelling imagery in a variety of ways. Google Deep Dream [17] and Neural Style Transfer [7] are recent examples demonstrating the power of deep neural networks. However, evolutionary computation has a long history in the field of computer generated art [5,8,10,18]. This paper combines Picbreeder [21], a system based on Compositional Pattern Producing Networks (CPPNs [23]), with Wave Function Collapse (WFC [9]), a procedural content generation method, to create images in the style of Zentangles, a meditative art form (Fig. 1). CPPNs are known for producing naturalistic images. Picbreeder interactively evolves CPPNs taking into account user preferences. Users choose images out of a population to evolve to the next generation. The procedural content generation algorithm WFC is named after a concept from quantum physics, but is a constraint satisfaction algorithm that arranges input tiles into a larger output pattern based on adjacency rules.

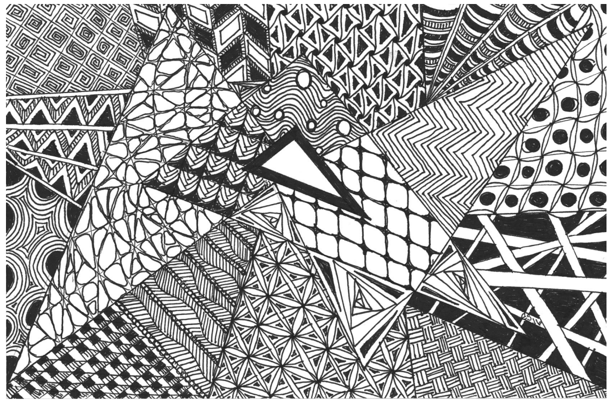


Fig. 1. Hand-drawn Zentangle. Image has sections with distinct tangle patterns. Art in this style is emulated by the system described in this paper. Credit to Elissa Schrum.

While Picbreeder alone can produce compelling images, it is limited by the activation functions in its CPPNs and the willingness of users to spend time evolving sufficiently intricate networks. WFC adds levels of complexity by introducing small rotational variation and repetition which Picbreeder would have trouble replicating on its own. The resulting output can appear more complex and makes better use of repetition than output produced by Picbreeder alone.

To create Zentangles, images are first evolved in Picbreeder. Images can be black and white (standard for Zentangles) or contain colors. Then the system takes some images, called tile images, and arranges them into patterns using WFC. Specifically, WFC generates rotations and reflections of the tiles, assigns random adjacency rules to them, and uses them to create pattern images. Finally, patterns are combined into a Zentangle according to a partitioning of the space based on one or two template images also taken from the evolved population.

Images for Zentangles can be generated by interactive or automated evolution. Interactively evolved Zentangles reflect the changing experience of a user over time and a user's aesthetic preferences. The user is directly involved in making Zentangles, though randomness influences the outcomes and adds an element of surprise. To automatically evolve Zentangles, the multi-objective evolutionary algorithm NSGA-II [6] was used with a variety of fitness functions. Automatic generation creates a lineage of art from which a user can pick their favorites.

The automatically generated Zentangles prove that human input is not required for the system to produce visually stimulating images. However, the automated system does produce degenerate output on occasion. In contrast, interactively generated Zentangles are based on a user's preferences, making the process more enjoyable to a user and the output possibly more rewarding.

2 Related Work

Computer generated art is a rich and diverse field. Evolution based approaches have a long history in the field [5,8,18]. The approach in this paper is an extension of Picbreeder [21], which uses Compositional Pattern Producing Networks (CPPNs [23]) to generate 2D images. The Picbreeder output is then combined with Wave Function Collapse (WFC [9]) to create Zentangles.

2.1 Art via Compositional Pattern Producing Networks

CPPNs [23] are a generative encoding capable of representing images [21], animations [26], 3D sculptures [4], neural networks [24], and soft-body robots [3]. They are arbitrary topology neural networks evolved via NeuroEvolution of Augmenting Topologies (NEAT [25]) to create images with principles of design seen in nature, such as repetition, repetition with variation, and symmetry. These abilities come from the activation functions CPPNs contain, such as symmetric, asymmetric, and periodic functions. CPPNs work by being repeatedly queried with coordinates in some geometric space. For example, querying x-coordinate values from -1 to 1 and passing them through a Gaussian function results in a symmetric pattern. A full list of available activation functions is in Section 3.1.

An early demonstration of CPPNs evolving images was Picbreeder [21], an interactive evolution system. Picbreeder has been extended in many ways. Notably, Endless Forms [4] uses CPPNs to evolve 3D forms made of voxels. Both 2D and 3D animations were generated with AnimationBreeder and 3DAnimationBreeder by adding a time input to CPPNs [26]. A time input has also been used to evolve audio timbres using Breedesizer [13]. CPPNs have also been used in the Procedural Content Generation video games Infinite Art Gallery [12], Artefacts [19], and Petalz [20]. Finally, DrawCompileEvolve uses human-crafted images to initialize a CPPN population, which is then evolved with Picbreeder [28].

2.2 Procedural Content Generation with Wave Function Collapse

Wave Function Collapse (WFC [9]) generates images from a set of tile images by repeating them in a coherent way that takes into account adjacency rules associated with each tile. Named after the phenomenon in quantum physics, WFC is essentially a constraint satisfaction algorithm [15]. Adjacency constraints determine which tiles are allowed to be next to each other, and in which orientation. Tiles are placed probabilistically according to the *minimum entropy* heuristic, which favors placements in locations with fewer available options.

WFC has been used for level generation in the games Proc Skater[1] and Caves of Qud[2]. WFC can also be extended for use on 3D objects and meshes, as well as anything that can be represented by following strict constraints, as shown through Martin O'Leary's WFC poetry[3]. O'Leary used syllables as "tiles" and poetic devices as constraints [15]. In creating Zentangles, WFC is a tool to create patterns out of simple tiles that are later placed into a larger composition.

[1] `https://arcadia-clojure.itch.io/proc-skater-2016`
[2] `http://www.cavesofqud.com/`
[3] `https://libraries.io/github/mewo2/oisin`

2.3 Zentangle

Zentangle is a meditation-based art form[4]. Creating Zentangles is a mindful, meditative process similar to doodling while practicing Zen meditation [16]. A Zentangle consists of a large shape filled with sections of free-hand patterns called tangles (Fig. 1). Tangles are drawn deliberately and thoughtfully, yet unconcerned with realism or traditional definitions of artistic skill. Zentangle mixes meditation and art in a way that is meant to transcend skill insecurities and foster creativity and introspection. However, because the end results have their own artistic merit, it is interesting to procedurally generate Zentangles, and ignore the meditative aspect of the process.

Computer generated images that look similar include fractal images, reaction-diffusion images [22], and Islamic star patterns [14]. Fractals produce patterns reminiscent of snowflakes, frost, and romanesco broccoli. Bacteria and brain coral growth patterns arise from reaction-diffusion models. Islamic star patterns are rigid and mathematical, an ode to the tile mosaics that the patterns are found in. All of the algorithms produce repetitive yet naturalistic results.

Fractals and Islamic star patterns are intricate, but too repetitive to represent the variation in Zentangles. Reaction-diffusion images have more variety, but still create fairly uniform patterns. The approach to Zentangles in this paper allows for a compelling mix of repetition and variation, both due to the expressive power of CPPNs, and the variety of ways WFC can combine tiles into a pattern.

3 Methods

A Zentangle is made by evolving images using CPPNs, composing them into patterns with WFC, and combining patterns into a final Zentangle. Image evolution can be interactive or automated. All source code is available for download at `https://github.com/sarahkfriday/QuantumZentanglement`.

3.1 Compositional Pattern Producing Networks

CPPNs have arbitrary topologies and are evolved using NeuroEvolution of Augmenting Topologies (NEAT [25]), which increases the complexity of neural network architectures over generations. The increase in complexity hinges on NEAT's use of structural mutation operators: new neurons can be spliced along existing links, and new links can form between existing neurons. Because CPPNs contain a variety of distinct activation functions, each new neuron has a randomly selected activation function, although there is also a mutation operation that changes the activation function in an existing neuron. Link weights can also be perturbed via mutation, and the use of historical markings for all new components allows crossover to align components with shared ancestry.

Activation functions within the CPPN's arbitrary topology produce the patterns. The full list of activation functions available are sigmoid, hyperbolic tangent, identity, Gaussian, sine, absolute value, linear piecewise, sawtooth wave,

[4] `https://zentangle.com/`

ReLU, softplus, triangle wave, square wave, cosine, and SiL. For many of these functions, multiple versions are available, such as *half* versions whose range is $[0, 1]$ and *full* versions whose range is $[-1, 1]$. Some of these activation functions are simply present because the Picbreeder code used [26] was originally part of a larger, more general neuroevolution system called MM-NEAT[5]. Although use of all of these functions is not necessarily recommended, all are available to humans when interactively evolving images and the specific set of functions used can result in large qualitative differences in the images generated.

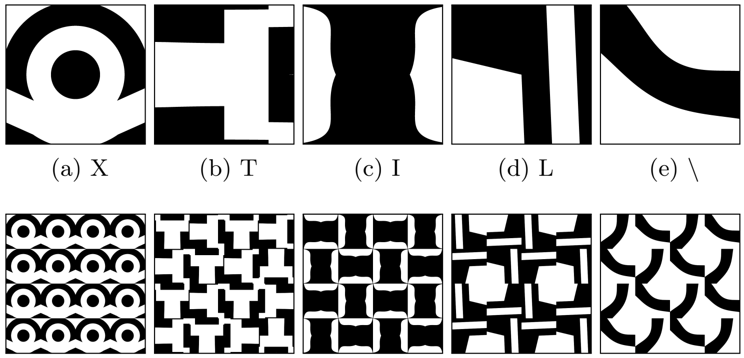
CPPNs can function on n-dimensions given n orthogonal inputs [21] as shown through some of Picbreeder's extensions. However, Zentangles are strictly two dimensional. The CPPN is given x and y inputs on a [-1, 1] scale that correspond with every pixel in the image, as well as a d input which represents the distance from the center [21]. Utilizing this tertiary input allows the CPPN to easily create radial patterns. CPPNs encode patterns at infinite resolution, because they can be queried on arbitrarily dense coordinate frames, but the output images in this paper are 1440×1440 pixels. The x, y, and d inputs associated with each pixel are input to the CPPN to determine the color of that pixel.

As in the original Picbreeder, CPPN outputs encode a color using hue, saturation, and brightness. This paper introduces a black and white only option which is different from the grayscale option from the original Picbreeder [21] because *only* black and white are present. These black and white images are produced by using a saturation value of 0, rendering hue irrelevant, and adjusting the brightness values. First, the minimum and maximum CPPN brightness outputs across all pixel coordinates are calculated, in order to calculate the midpoint between them. Next, the actual brightness associated with each pixel is 1 (white) if the original brightness is above the midpoint, and 0 (black) otherwise. The results are dual-toned black and white images, which is the style of traditional Zentangles. In a departure from traditional Zentangles, the system can also create color Zentangle images. To enable switching back and forth between color and black-and-white images during interactive evolution, the hue and saturation outputs of CPPNs are always maintained. However, it is still necessary for template images to have large black regions, so that they can partition the space according to black and non-black regions (see Section 3.3). For this reason, only the areas that would be white in a black and white image may contain colors.

3.2 Wave Function Collapse

Wave Function Collapse is a set of two algorithms, Simple-Tiled and Overlapping, that solve the adjacency constraint problem [9]. Only Simple-Tiled is used in this paper, and it is thus the only algorithm discussed. The algorithm first reads a list of square image tiles and their adjacency constraints. Next, an output array is initialized with each index representing a tile. This array is known as the wave. Elements in the wave are true or false, indicating if the tile is forbidden at that position in the output. The wave is initialized as unobserved, meaning every

[5] `https://github.com/schrum2/MM-NEAT`



(a) X (b) T (c) I (d) L (e) \

(f) X Pattern (g) T Pattern (h) I Pattern (i) L Pattern (j) \ Pattern

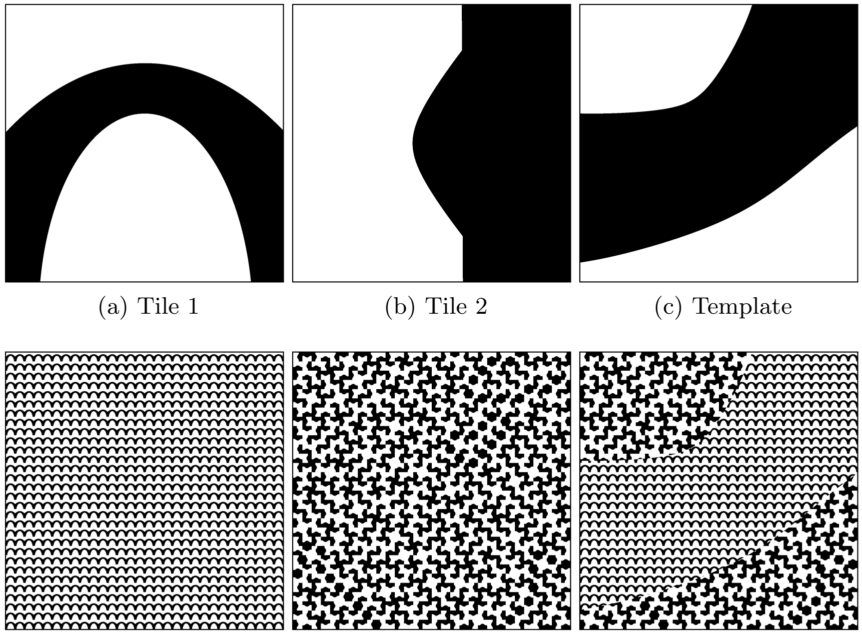
Fig. 2. Pattern image creation with WFC using black-and-white tiles evolved by Picbreeder. The tile images have been assigned the X, T, I, L and \ symmetry types. Below each tile image is the pattern image created by WFC using the assigned symmetry type. Each type creates specific rotations and reflections. When multiple tiles are present in one pattern, each can have its own symmetry type (not shown).

element is true. Thus begins observation of the wave function. The algorithm selects a tile with the shortest non-empty list of adjacency constraints (lowest nonzero entropy) for that given position in the output. Once selection is final, the tile becomes observed and its information collapsed into the wave by propagation. The algorithm repeats the observation phase until the entire wave is observed returning an output, or a contradiction arises that cannot be resolved.

In Zentangle creation, Picbreeder images are tiles for the Simple-Tiled algorithm. Tiles are rendered by CPPNs at a resolution of 48×48 pixels, then randomly assigned symmetry types named after letters associated with symmetry patterns isomorphic to certain characters as shown in Fig. 2. Each symmetry type generates particular rotations and reflections of the tile image, and indicates how those rotations can be placed adjacent to other rotations of the image, or other images. A tile set can consist of rotations and reflections of one or more Picbreeder tiles. Each tile set produces one background pattern via WFC on a 30×30 grid, which produces a 1440×1440 pixel image. Once two or more pattern images are created from all tile sets, they are combined into a Zentangle.

3.3 Creating Zentangles

Details of Zentangle formation depend on how many images contribute to the Zentangle. The most straightforward example uses three images (Fig. 3). Two are randomly selected as tile images. Each tile is assigned a random symmetry type and creates a pattern via WFC, as described above. The third image is a template, which splits different regions of the Zentangle. The Zentangle is created by analyzing each coordinate location of the template image. If the template is black at that coordinate, it is replaced with the color at that same coordinate from one of the pattern images. If the template is not black at that coordinate, it is replaced with the color at that same coordinate from the other pattern image.



(a) Tile 1 (b) Tile 2 (c) Template

(d) Pattern Image 1 (e) Pattern Image 2 (f) Final Zentangle

Fig. 3. Zentangle creation from three images. The two images (a) and (b) are randomly assigned to be tile images, so the third image (c) becomes the template. Wave Function Collapse generates pattern image 1 (d) from tile 1 (a) and pattern image 2 (e) from tile 2 (b). The final Zentangle (f) is created by assigning pattern 1 to the black areas of the template image, and pattern 2 to the non-black areas of the template image.

The template therefore defines a partitioning of the space in terms of black and non-black with one pattern being mapped to each partition.

Zentangles can also be created with other input image counts (Table 1). In some cases, one image is used as both a tile and a template. If two images are used as templates, then different patterns occupy three regions: black areas in both templates, non-black areas in both templates, and areas black in one but non-black in the other. When six or more images are selected, then individual background patterns consist of multiple tiles, each with their own random symmetry type. Excess tiles are split evenly across two background patterns.

3.4 Interactive Evolution via Selective Breeding

One way to generate images for Zentangles is interactively, as done in Picbreeder. This selective breeding algorithm uses pure elitist selection. First the user sees $N = 20$ images, then selects $M < N$ individuals as parents for the next generation. These parents are also directly copied to the next generation. Mutation and crossover operations from NEAT are used to create offspring CPPNs. Selection continues in this fashion for as long as the user likes. There is a 50% chance of crossover for each offspring. Whether an offspring has two parents or is a clone,

Table 1. Zentangle Construction. When *Total* number of images are selected, some are used as *Templates* and some are used as *Tiles*. In some cases, one image is used as both types. Details in *Additional Information*.

Total	*Templates*	*Tiles*	*Additional Information*
2	1	2	One image used as both template and tile.
3	1	2	Template and tile images are distinct.
4	2	3	One image used as both template and tile. Uses intersection of two templates. Three distinct background patterns.
5	2	3	Template and tile images are distinct. Uses the intersection of two templates. Three distinct background patterns.
6+	1	5+	Multiple images are used in each of two background patterns.

it then has a certain number of mutation chances defined by a user-controlled slider ranging from 1 to 10. For each mutation chance, these rates apply: 30% activation function change rate, 5% per-link weight perturbation rate, 40% link creation rate, and 20% node splice rate. These parameters are the same ones used in the base Picbreeder code [26]. The user also has the option at any point to select images and create a Zentangle out of them, as described above.

3.5 Automated Evolution with NSGA-II

Automated experiments used the multi-objective evolutionary algorithm Non-dominated Sorting Genetic Algorithm II (NSGA-II [6]) to evolve images. NSGA-II uses $(\mu + \lambda)$ selection, specifically $\mu = \lambda = 16$ in this paper, to create new parent populations of size μ from combined parent and child populations of size $\mu + \lambda$. Each child population is created by performing selection on the parent population, and applying crossover and mutation at the same rates used in the interactive experiments (only one mutation chance per offspring). After creating children, the algorithm sorts the combined parent/child population into Pareto layers according to the multi-objective dominance relation, by which one solution is superior to another if it is at least tied in every objective and strictly better in at least one. The first layer is the Pareto front of the population, meaning that it contains no dominated solutions. Removing layers reveals the Pareto fronts of the remaining population members. Elitist selection favors individuals in the less dominated layers, and within layers selection favors solutions that are more distinct from others in their layer in terms of fitness, as determined by a crowding distance metric. Using the crowding distance metric ensures that individuals are evenly spread across the trade-off surface between objectives.

Of the activation functions in Section 3.1, the ones used in automated experiments were full sigmoid, full Gaussian, cosine, sine, identity, and half linear piecewise, which were selected because they provide representation of symmetric, periodic, and asymmetric functions. However, other functions could also be used to produce interesting results. For each parent and child population, three Zentangles are created. For each Zentangle, a number between two and six was randomly generated to determine the number of images selected from the population. Those images are then used to compose the Zentangle.

Three different selection schemes were used to evolve images: Random, Half-Black, and Half-Black-3-Colors. The Random scheme selects random images at each generation. The Half-Black scheme favors images whose ratio of black to white pixels is as close to 0.5 as possible. The exact fitness to maximize is $hb(\cdot)$:

$$hb(p) = -\left|\frac{\sum_x \sum_y black(p_{x,y})}{width(p) \times height(p)} - 0.5\right| \tag{1}$$

where p is a picture, $p_{x,y}$ is the color of a pixel at coordinates (x, y), and $black(p_{x,y})$ is 1 if the designated pixel is black, and 0 otherwise. The Half-Black-3-Colors scheme has four fitness functions: the Half-Black fitness function, and one additional function for each of the RGB color channels. Specifically, the fitness functions for the three channels $red(\cdot)$, $green(\cdot)$, and $blue(\cdot)$ are:

$$red(p) = \frac{sum(p, red) - sum(p, blue) - sum(p, green)}{((width(p) \times height(p)) - \sum_x \sum_y black(p_{x,y})) + 0.0001} \tag{2}$$

$$blue(p) = \frac{sum(p, blue) - sum(p, red) - sum(p, green)}{((width(p) \times height(p)) - \sum_x \sum_y black(p_{x,y})) + 0.0001} \tag{3}$$

$$green(p) = \frac{sum(p, green) - sum(p, blue) - sum(p, red)}{((width(p) \times height(p)) - \sum_x \sum_y black(p_{x,y})) + 0.0001} \tag{4}$$

$$sum(p, c) = \sum_x \sum_y intensity(p_{x,y}, c) \tag{5}$$

where the $sum(p, c)$ for a given color channel c is the sum of the pixel intensity values of that channel across all pixels in p. The *intensity* function returns the color value in the range $[0, 255]$ of the pixel $p_{x,y}$ for channel c. The value 0.0001 in the denominators is a small term to prevent division by 0. The denominators scale values with respect to the number of non-black pixels to assure that lack of color in the black areas is not punished. Sums from other color channels are subtracted so that each objective is in conflict with the others, encouraging a diversity of colors. These fitness functions do not attempt to evolve to a particular best result, but trajectories where results along the way can be analyzed by a user. Evolutionary runs with each selection scheme were conducted for 50 generations.

4 Results

Both interactively and automatically generated art are discussed individually. All results can be viewed at `southwestern.edu/~schrum2/re/zentangle.php`.

4.1 Interactively Generated Art

The authors made extensive use of the interactive system. When generating art by selecting images, users maintain influence on the generated Zentangles. Users decide which activation functions are available, and how many mutation chances each offspring has. Although users cannot control the symmetry types assigned to tiles or choose which images become templates, they can control the complexity of Zentangles by deciding which images to include, and how many. Users can also repeatedly generate new Zentangles with the same images, and rely on randomness in the generation process to yield a range of different results.

Users also learn how to generate art that is most appealing to them. By seeing how their selections are used to generate art, they develop an understanding of which image combinations produce what is, in their opinion, the "best" art.

Table 2 shows examples of interactively generated art in black-and-white and color derived from different numbers of images. It demonstrates the variety of results produced by human users. The strict borders between patterns creates a tense interaction between patterns, competing for dominance in pictorial space. The black and white Zentangles contain a depth of space chiseled out by the luminosity of the patterns. Those patterns that are darker recede while those that are lighter advance, dancing through the image. Color Zentangles have a rich balance of complementary and analogous colors which further contrast the individual patterns. Zentangles composed of four to five images entwine patterns in an intricate performance, deepening the pictorial space. Zentangles of six images can be quite noisy, like the black-and-white examples of Table 2. Color examples with six images can also be noisy, but the examples in Table 2 carve out distinct color regions, which gives them distinct forms despite the noise.

Selecting which activation functions can be added to CPPNs results in qualitatively different Zentangles (Table 3). ReLU leads to sharp corners, softplus and Gaussian produce round edges and circles, sawtooth wave creates choppy patterns, and sine with cosine leads to wobbly curves. The user's preferred aesthetic is thus reflected in the end product, making for an enjoyable process.

4.2 Automatically Generated Art

While interactive generation of Zentangles yields appealing results, similar results can also be generated automatically. Table 4 contains examples of automatically generated Zentangles similar in quality to interactively generated Zentangles. There is a large range of patterns and colors.

The Random scheme produced these results despite not selecting for any particular features. Images had a wide range of patterns and colors, but occasionally led to homogeneous degenerate Zentangles (Table 5). Zentangles using distinct but similar background tiles can create images where boundaries between patterns are unclear. Because the population is small, distinct offspring may have the same parents and therefore look similar. Still, random selection maintains diversity across generations overall, since it is not honing in on a particular goal.

The Half-Black scheme is meant to assure a good mix of black and non-black regions in each image, since template images depend on this distinction. However, this fitness scheme exerted a high selection pressure, resulting in highly converged populations, leading to more homogeneous degenerates. This kind of convergence is not uncommon in evolutionary art techniques [10,27]. Because there is a specific fitness objective that can be optimized, optimal scoring images

Table 2. Interactively Generated Zentangle Images. Each row indicates the number of user-selected images that were used to generate the Zentangle. The first two columns show black-and-white images, and the next two columns show color images. This is a small sampling of the range of images humans can generate.

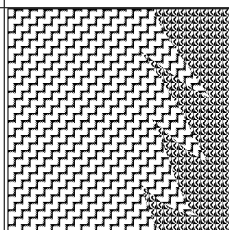
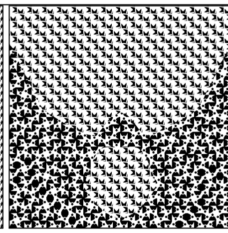
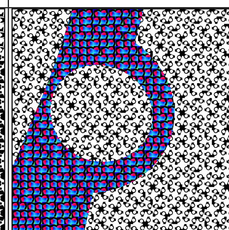
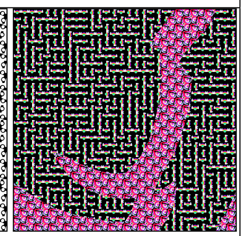
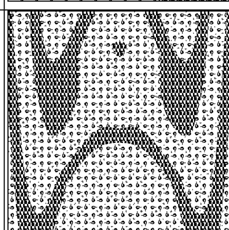
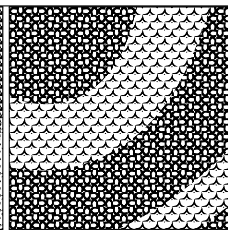
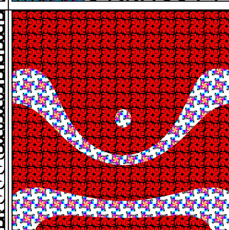
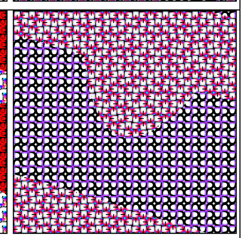
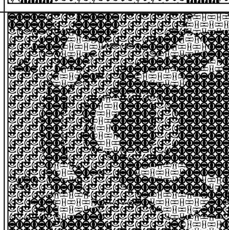
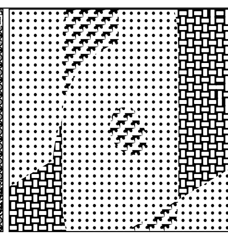
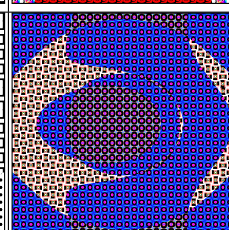
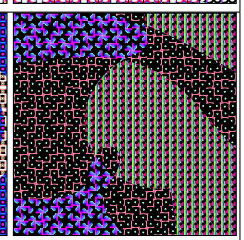
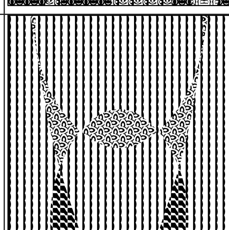
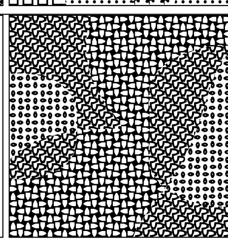
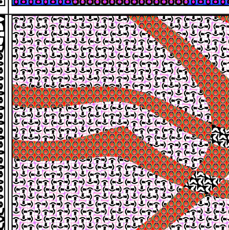
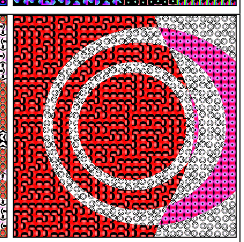
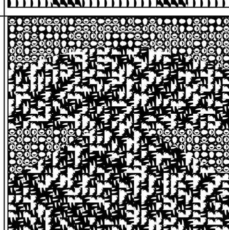
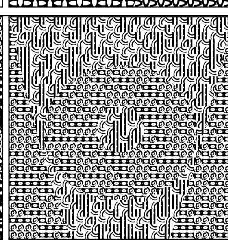
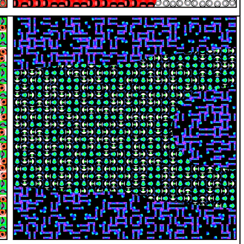
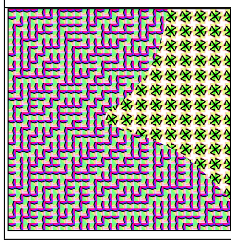
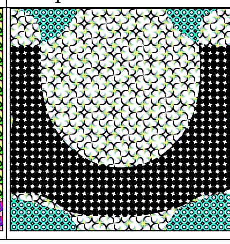
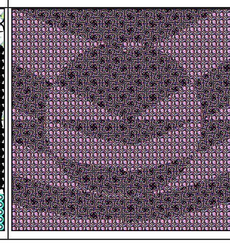
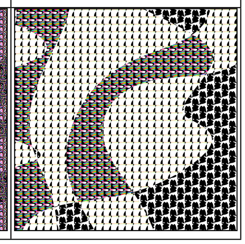
#	Black and White		Color	
2				
3				
4				
5				
6				

Table 3. Zentangle Images Evolved by Humans with Specific Activation Functions. Each column indicates the only available activation functions used to generate images for the Zentangle, demonstrating qualitative distinctions between activation functions.

ReLU	Softplus & Gaussian	Sawtooth Wave	Sin & Cos
			

quickly take over the population when they emerge. It is apparently easy for CPPNs to generate images with a black bar covering the lower half of the image (Table 5), which is a simplistic pattern. However, WFC's random symmetry assignments can at least make the background patterns interesting in some cases.

The Half-Black-3-Color scheme builds upon the Half-Black scheme by encouraging a variety of color combinations in the white regions of images. This decision maintains more diversity throughout the generations because colors are in competition, and the population is too small to maintain a proper Pareto front across all four objectives. Therefore, this approach keeps producing unexpected designs throughout the entire 50 generation run.

5 Discussion and Future Work

The Zentangle system is able to produce a diverse range of vibrant and interesting images reminiscent of Zentangles. The interactive system shares Picbreeder's ability to adapt to a user's change in aesthetic preferences as the experience progresses. As users evolve images, they may discover their affinity for a curvilinear composition, or a symmetric one. The ability to choose the activation functions for the CPPNs allows for the expression of such preferences. Users also control the complexity level of the resulting Zentangle images via their image choices.

Automated generation is capable of creating similar Zentangles without any human input. However, the automated process occasionally yields degenerate results. Specifically, automated generation can converge to a population of homogeneous images in some cases, as happened with the Half-Black fitness function. Random selection of input images may even pick out similar images from an otherwise diverse population. When multiple background patterns are made from similar tiles, the results are sometimes simplistic.

Some Zentangles composed of six images demonstrate that combining many distinct tiles in one pattern can create a cacophony of line and color. Some of the chaos can be attributed to randomness in assigning symmetry types and image roles. However, appealing compositions sometimes emerge out of the chaos.

Better fitness functions could improve the automatic generation of Zentangles. Fitness functions inspired by Birkhoff's complexity and order measure [2]

Table 4. Automatically Generated Zentangle Images. Each row indicates the number of images that were used to generate the Zentangle. Each column indicates the fitness scheme which yielded the Zentangle. These automatically generated results are comparable to those generated via interactive evolution.

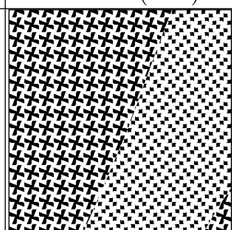
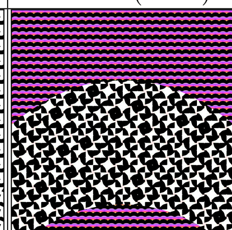
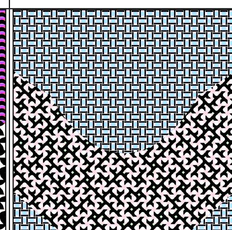
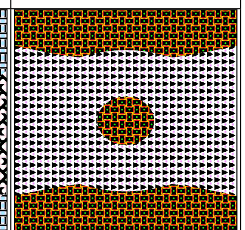
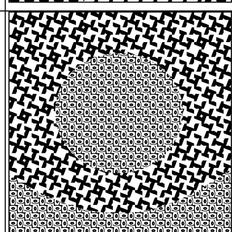
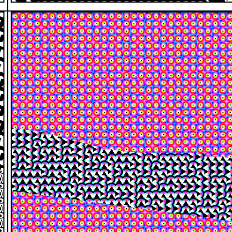
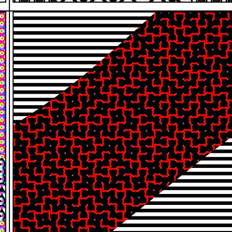
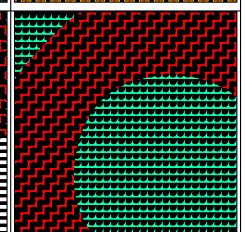
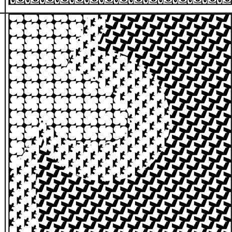
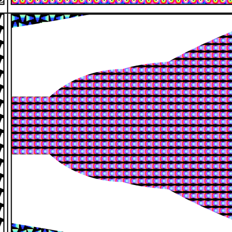
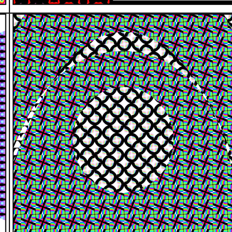
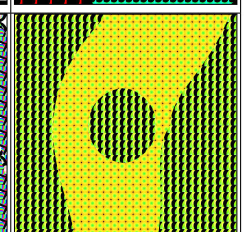
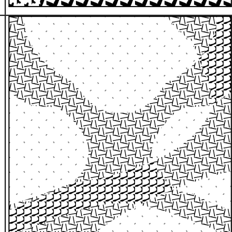
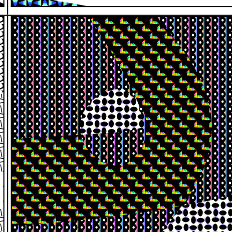
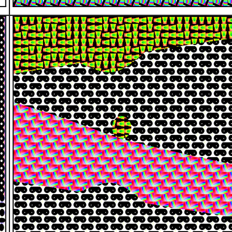
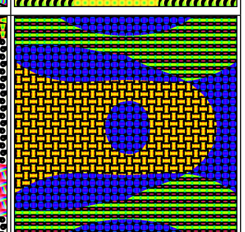
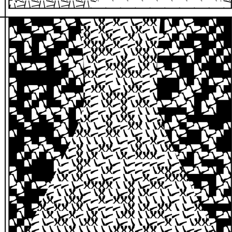
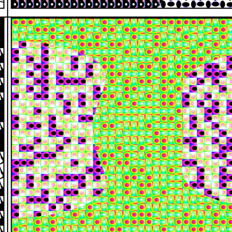
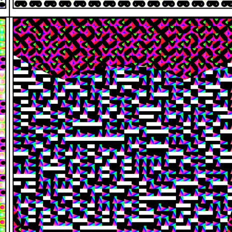
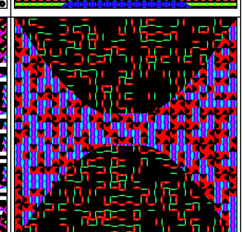
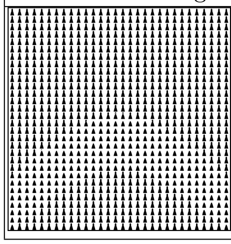
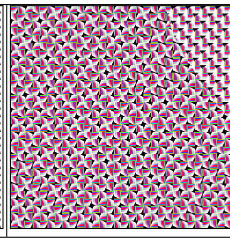
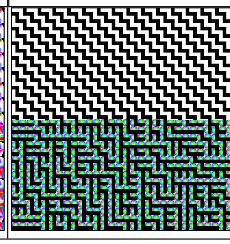
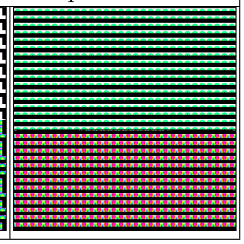
#	Random (BW)	Random (Color)	Half-Black	Half-Black-3-Color
2				
3				
4				
5				
6				

Table 5. Automatically Generated Degenerate Zentangles. When nearly identical tiles are used to create all background images, the result is a homogeneous pattern, as shown in the first two images. The next two images demonstrate a problem with the Half-Black fitness function, which is easy to optimize: all population members have a black band around one half of the image, though the color portion exhibits variation.

Homogeneous Tiles		Half-Black Template	
			

as well as more recent aesthetic measures could produce a better population of images to create Zentangles from. Basic aesthetic principles worth evaluating include symmetry, repetition, rhythm, and contrast [11], which are elements of design well known within the art community [1]. Heijer and Eiben also proposed several aesthetic measures [10] which could be useful as objectives in multi-objective evolution of images for automated Zentangle creation.

After 50 generations of automatically generating Zentangles, collections of appealing images were obtained with each approach, but results contain several degenerate cases. Half-Black fitness leads to the most homogeneous output, with more diverse images occurring in earlier generations. Random fitness has degenerates sprinkled throughout the generations, but generally maintains high quality and diversity. The Half-Black-3-Color scheme also leads to high-quality results, with images that are even more often filled with diverse colors. Still, it could be beneficial to intelligently select template images from the population. Even in interactive runs specifying the template image would be nice.

Additionally, intelligently configuring WFC may create more structured patterns. For example, some tiles may lend themselves to certain symmetry types. The ability to intelligently associate tiles with symmetry types could create patterns with improved balance, rhythm, and repetition. Alternatively, the user could select the symmetry type for each tile. Of course, demanding too much input from the user could make the system tedious to use and prevent the serendipitous discovery of unexpected output. Similarly, although it may be possible to define criteria for which images to evolve and how to combine them into Zentangles, doing so successfully may take more effort than it is worth since the results become more restricted as the specificity of the fitness functions grows. The results from the Random fitness function are already interesting and pleasing in most cases, so increasing the expressive power of the system may be more valuable than strictly controlling the selection process.

There remain a number of differences between hand-drawn Zentangles and the Zentangles produced in this paper. One difference is variation in pattern scale that can be found in hand-drawn Zentangles. The current system lacks scale variation, but this feature should be easy to add because CPPNs can render images at arbitrary resolutions [23]. Hand-drawn Zentangles can also have patterns in arbitrary orientations, which is another enhancement that should be easy to add. Allowing for more than three background patterns is yet another feature that would make generated results similar to hand-drawn ones. Perhaps the most challenging types of patterns from hand-drawn Zentangles for this system to produce are organic patterns not produced by repetition of tile-like elements. However, a hybrid system that incorporates art from other systems, such as the reaction-diffusion images mentioned earlier [22], could provide the occasional organic pattern. Using WFC's Overlapping algorithm instead of the Simple-Tiled algorithm also has potential to make more organic patterns [9].

Whether the generated Zentangles are less visually appealing than hand-drawn Zentangles is a matter of subjectivity. However, the proposed enhancements should at least produce results closer to what humans can draw.

6 Conclusion

Through the combined use of Picbreeder and Wave Function Collapse, a computer can generate complex, Zentangle-esque art. In fact, a human user is not even needed to produce the images, though human input can be valuable, since humans recognize which images might combine to produce the best results. Automatic generation of Zentangles utilizing three different fitness functions has proven its ability to produce intriguing results as well, despite some degenerate output. However, further development should strive to close the qualitative gap between evolved and hand-drawn Zentangles, by allowing for more diverse pattern types and ways of combining them. Such progress may arise from combining multiple techniques into hybrid systems, as was done in this paper.

References

1. Arnheim, R.: Art and Visual Perception: A Psychology of the Creative Eye. University of California Press (1954)
2. Birkhoff, G.: Aesthetic Measure. Harvard University Press (1933)
3. Cheney, N., MacCurdy, R., Clune, J., Lipson, H.: Unshackling Evolution: Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding. In: Genetic and Evolutionary Computation Conference. ACM (2013)
4. Clune, J., Lipson, H.: Evolving Three-dimensional Objects with a Generative Encoding Inspired by Developmental Biology. In: European Conference on Artificial Life. pp. 141–148 (2011)
5. Dawkins, R.: The Blind Watchmaker. Longman Scientific and Technical (1986)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**, 182–197 (2002)
7. Gatys, L., Ecker, A., Bethge, M.: A Neural Algorithm of Artistic Style. Journal of Vision **16**(12), 326 (2016)
8. Greenfield, G.: Robot Paintings Evolved Using Simulated Robots. In: Applications of Evolutionary Computing. pp. 611–621. Springer, Berlin, Heidelberg (2006)

9. Gumin, M.: WaveFunctionCollapse. GitHub repository (2016), `https://github.com/mxgmn/WaveFunctionCollapse`
10. den Heijer, E., Eiben, A.E.: Comparing Aesthetic Measures for Evolutionary Art. In: European Conference on the Applications of Evolutionary Computation. pp. 311–320. Springer (2010)
11. Hoenig, F.: Defining Computational Aesthetics. Computational Aesthetics in Graphics, Visualization and Imaging pp. 13–18 (2005)
12. Hollingsworth, B., Schrum, J.: Infinite Art Gallery: A Game World of Interactively Evolved Artwork. In: IEEE Congress on Evolutionary Computation (2019)
13. Jónsson, B.T., Hoover, A.K., Risi, S.: Interactively Evolving Compositional Sound Synthesis Networks. In: Genetic and Evolutionary Computation Conference. pp. 321–328. ACM (2015)
14. Kaplan, C.S.: Computer Generated Islamic Star Patterns. In: Bridges 2000: Mathematical Connections in Art, Music and Science. pp. 105–112 (2000)
15. Karth, I., Smith, A.M.: WaveFunctionCollapse is Constraint Solving in the Wild. In: Foundations of Digital Games. ACM (2017)
16. Kopeschny, D.A.: The Phenomenological Experience of Zentangle and the Implications for Art Therapy. Master's thesis, St. Stephens College, Alberta, Canada (2016)
17. Mordvintsev, A., Olah, C., Tyka, M.: Inceptionism: Going Deeper Into Neural Networks (2015), `https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`
18. Neumann, A., Szpak, Z.L., Chojnacki, W., Neumann, F.: Evolutionary Image Composition Using Feature Covariance Matrices. In: Genetic and Evolutionary Computation Conference. ACM (2017)
19. Patrascu, C., Risi, S.: Artefacts: Minecraft Meets Collaborative Interactive Evolution. In: Computational Intelligence and Games. pp. 349–356. IEEE (2016)
20. Risi, S., Lehman, J., D'Ambrosio, D.B., Hall, R., Stanley, K.O.: Petalz: Search-based Procedural Content Generation for the Casual Gamer. IEEE Transactions on Computational Intelligence and AI in Games (2015)
21. Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., Stanley, K.O.: Picbreeder: A Case Study in Collaborative Evolutionary Exploration of Design Space. Evolutionary Computation **19**(3) (2011)
22. Sims, K.: Interactive Evolution of Dynamical Systems. In: European Conference on Artificial Life. pp. 171–178 (1992)
23. Stanley, K.O.: Compositional Pattern Producing Networks: A Novel Abstraction of Development. Genetic Programming and Evolvable Machines **8**(2), 131–162 (2007)
24. Stanley, K.O., D'Ambrosio, D.B., Gauci, J.: A Hypercube-based Encoding for Evolving Large-scale Neural Networks. Artificial Life (2009)
25. Stanley, K.O., Miikkulainen, R.: Evolving Neural Networks Through Augmenting Topologies. Evolutionary Computation (2002)
26. Tweraser, I., Gillespie, L.E., Schrum, J.: Querying Across Time to Interactively Evolve Animations. In: Genetic and Evolutionary Computation Conference (2018)
27. Woolley, B.G., Stanley, K.O.: On the Deleterious Effects of a Priori Objectives on Evolution and Representation. In: Genetic and Evolutionary Computation Conference. pp. 957–964. ACM (2011)
28. Zhang, J., Taarnby, R., Liapis, A., Risi, S.: DrawCompileEvolve: Sparking Interactive Evolutionary Art with Human Creations. In: Evolutionary and Biologically Inspired Music and Art. pp. 261–273. Springer (2015)