Color in Nature Inspired Design

RANDY FAN, University of California, Berkeley
KENNY CHEN, University of California, Berkeley

ACM Reference Format:
Randy Fan and Kenny Chen. 2019. Color in Nature Inspired Design. 1, 1 (December 2019), 5 pages.

1 INTRODUCTION

Cellular automaton (CA) and reaction-diffusion (RD) systems are both mathematical models capable of computing complex state transitions and simulating natural and physical phenomena (hence, our usage of the phrase "Nature Inspired Design"). For example, given the elementary CA Rule 30, a cellular automaton has the capability to generate a pattern that resembles a Conus Textile snail's shell.

Although CA and RD models have been studied quite extensively, the colors used in each iteration of the algorithms have not been experimented with as often. In the simplest case of binary automaton, a cell is most commonly colored white (alive) or black (dead), resulting in rather dull, bicolored generations. In plain reaction-diffusion models, the instances are often greyscale.

In our project, we applied certain color schemes to CA and RD instances primarily for data visualization and aesthetic purposes. For CA, we experimented with 3D totalistic cellular automata rules and applied certain color palettes to extract additional insights (e.g. of generations a cell has been alive). Specifically, we developed color palettes using the k-means algorithm and the idea of equidistant colors, as well as monochromatic, pastel, and similar hue color palettes. For both the CA and RD models, we leveraged their biologically plausible mechanism, combining the generated instances with smart color palettes to create realistic looking skin pattern. In other words, we define and created a new, simple procedure to generate textures using CA and RD rules. These seemingly distinct tasks all share the common goal of developing creative color palettes and applying them to CA and RD models in meaningful ways.

2 TOTALISTIC 3D CELLULAR AUTOMATA

We implemented a 3D outer totalistic CA starting from a 6x6x6 cube of live cells, where cell state at time t is defined by its state at time t-1 and the sum of the states of its neighbors at time t-1. At every generation, the grid cell becomes an outlined cube (alive) if the sum of the total number of cubes around it equals the number set by a rule.

- Our rules are defined the following way:
  - Example rule: [[4, 9], [5, 6, 3]]
    - 4, 9 = a cell needs to have 4 OR 9 live neighbors to continue living
    - 5, 6, 3 = a cell needs to have 5 OR 6 OR 3 live neighbors to be born (dead → live)

3 EQUIDISTANT COLORS

Equidistant color palette generation is useful in many cases, such as cross-referencing with a key in some data visualization, for example.

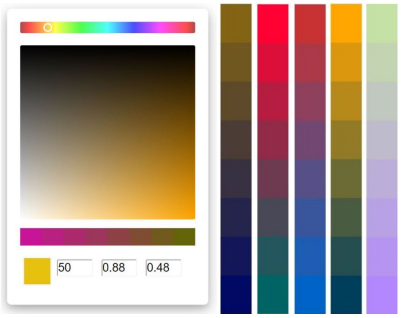

Fig. 1. HSL color-picker tool with visually equidistant color palette generation using color interpolation.

We use an interpolation-based algorithm to find perceptually equidistant colors. Given two colors, c_1 and c_2, and an integer n that denotes the number of unique colors, we generate n colors that are between c_1 and c_2 and are visually equidistant. We essentially are sampling points along the line formed by points c_1 and c_2, each with equal distance from each other. It is important that we make the two endpoint colors relatively distinct. For example, it's a bad idea to have endpoint colors that are both shades of blue because it will be harder to tell the colors apart:

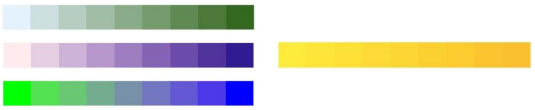

Fig. 2. Having the endpoint colors be similar leads to a palette in which the colors are difficult to discern.

We compared palettes generated using interpolation over HCL as well. We initially only interpolated over the RGB color space due to ease of implementation. Though many of the palettes generated seemed to have visually equidistant colors, many contained colors which were not perceptually equidistant.

Authors' addresses: Randy Fan, Department of Computer Science and Engineering, University of California, Berkeley; Kenny Chen, Department of Computer Science and Engineering, University of California, Berkeley.

XXXX-XXXX/2019/12-ART $15.00
https://doi.org/

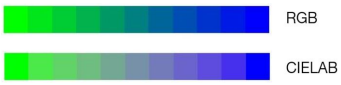

Fig. 3. It is difficult to discern the blues in the RGB palette.

As can be seen in the figure, it is difficult to tell the blues apart in the RGB palette. This is due to the fact that the RGB space is not perceptually uniform. We used a perceptually uniform color space, CIELAB, to solve this problem. Though interpolating over the CIELAB color space corrected our problems related to creating visually equidistant color palettes, in our application it was not necessarily important to make this distinction. This is because when the palette is applied to our CA, we found it more important to just distinguish whether cells had color closer to either of the endpoints.

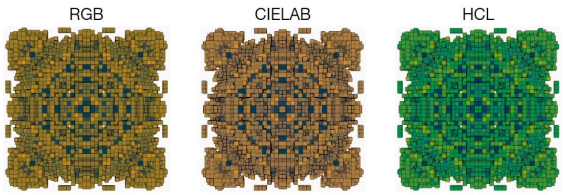

Fig. 4. Application of palettes created using interpolation over RGB, CIELAB, and HCL color spaces.

With the palettes generated using interpolation, we were easily able to answer questions like whether the CA rule favors cells born in earlier or later generations depending on whether there are more cells of color closer to one of the two endpoint colors.

4 K-MEANS ALGORITHM

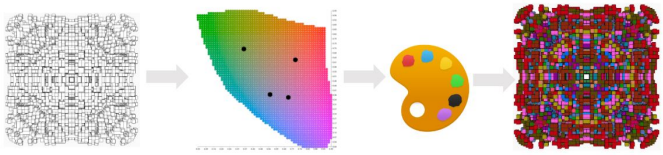

Fig. 5. Application of k-means palette.

To obtain unique, distinct colors for x generations, we used the k-means clustering algorithm to divide the CIELAB color space. Specifically, we sample the CIELAB color space and run k-means on these data points to get N clusters. From these clusters, we select the centroid of each as a color in our palette. Because the CIELAB color space has a wider range than the RGB color space, it was necessary to convert LAB values to the nearest RGB value because some of these LAB values do not have a corresponding RGB representation.

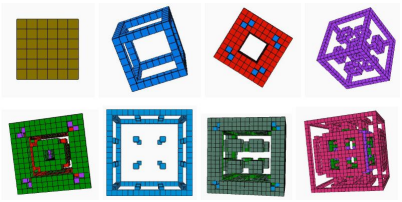

Fig. 6. First 8 generations using the K-means color palette and the rule [[1,2,3], [1,2,3]]

We found that there were two main applications of palettes created in this manner. The first was that it was easy to tell, at a glance, the number of cells which were born in different generations. The second application was that we could easily visualize the number of consecutive generations a cell remained alive. We were also able to constrain the CIELAB space while sampling data points for k-means to create palettes from a subspace of the CIELAB color space.

Fig. 7. First 8 generations using the K-means color palette and the rule [[1,2,3], [1,2,3]]

5 MONOCHROMATIC AND PASTEL

Creating palettes by constraining the CIELAB color space in this way was mainly performed with the goal of improving the aesthetics of the CA generated. This extension to our algorithm was done for generative artistic purposes and because adding color in this manner yields more insights and is more interesting than the usual black-white representation of these models.The sampled subspace also fulfills the same applications described previously using the k-means algorithm.

The monochromatic color palette returns a set of colors that vary in saturation and lightness, but have the exact same hue. So, in our algorithm we only sample points over the CIELAB color space with a hue equal to some predefined value.

A pastel color is defined to have high value and low to intermediate saturation [2]. We defined our pastel colors as having a saturation value between 15 and 40 and a value between 65 and 100. Fig. 6. Our pastel palette containing 8 colors Fig. 7. Generation 10 using the Pastel color palette and the rule [[1,2,6],[1,2,3]]

6 NATURE INSPIRED DESIGN: SKIN PATTERNS

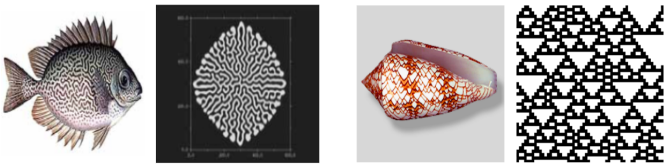

Fig. 8. Biologically-inspired designs.

The biologically plausible mechanisms of cellular automaton and reaction-diffusion systems can exhibit animal skin patterns that come from nonlinear dynamical microscopic systems of cell interactions [4]. As we researched and looked into specific examples of these generated skin instances, we noticed most examples were black and white or grayscale (shown in Figure 8). Thus, for the second part of our project, we extracted color schemes representative of real animal skin and input those colors into the of CA and RD algorithm used to mimic skin patterns.

Specifically, we implemented skin algorithms (one using the Gray-Scotts reaction-diffusion model and the other using probabilistic totalistic cellular automata) to simulate cheetah and ocellated lizard skin. We chose these specific animal skins because we were able to find accurate parameters that could realistically simulate those skins [4]

Our work on simulating animal skin reinforces the idea that cellular automaton and reaction-diffusion system are not just abstract computational systems, but capable of simulating natural and physical phenomenon quite realistically with the addition of color.

7 RGB BINNING

To add more realistic coloring to the generated animal skin instances, we needed to obtain color schemes representative of real animal skin. In order to do this, we developed a simple RGB binning approach to extract colors from images. We chose to implement RGB binning because it doesn't overrepresent dominant colors (as could be the case in K-means). For example, if the input image is a single color, the color will get grouped into one bin, while K-means would assign several clusters to that color. Also, RGB binning ensures details are captured in the result (e.g. a color that only takes up a small part of an image will still be located in a cube grid). However, one of the main disadvantages of RGB binning is that it could separately count clusters of pixels located near a boundary.

The main purpose of RGB binning in our project is to extract dominant colors from a real image of the relevant animal skin, and feed the color scheme into the specific animal skin algorithm.

Specifically, we first extract colors from a real image of animal skin. Then, we partition the image's RGB values into a uniform 3D grid. Next, we split the 3D grid into X cube grids and average the color values. Lastly, we return the top Y most dominant colors. Note X and Y are parameters we define.

The figure below displays the steps involved in obtaining the top 4 dominant colors from a real image of cheetah skin.

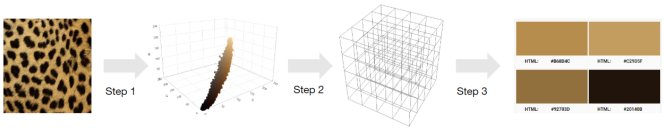

Fig. 9. RGB binning process.

8 GRAY-SCOTT MODEL

In order to simulate the cheetah skin pattern, we needed a robust reaction-diffusion model. We chose to use the classic Gray-Scott model because it has a surprising variety of interesting spatiotemporal patterns, often reminiscent of patterns occurring in nature.

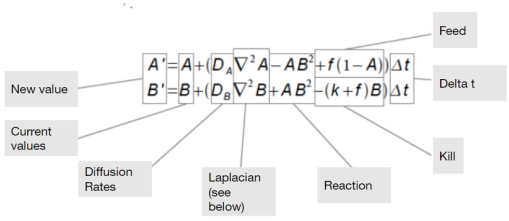

Fig. 10. Gray-Scott equations.

The Gray-Scott Model generates patterns that represent the spatial concentration of two chemicals, A and B (in our project, these chemicals can be thought of as skin pigmentation chemicals). The set of equations that define the update rule are shown in Figure 10. The Feed rate represents how much chemical A is being added to the system at every iteration. The Kill rate represents how much chemical B is being removed from the system at every iteration. The Reaction rate represents the likelihood chemical A will unite with two chemical B. If A is connected to two B's, chemical A would transform into B. The Diffusion and Laplacian functions modify the concentration of chemical A or B by introducing neighborhood effects (convolution) and simulating chemical diffusion.

9 CHEETAH SKIN PATTERN

We first researched parameters values that could realistically generate cheetah skin pattern. We decided to use Karl Sim's provided Gray-Scott parameters for a "mitosis" simulation, where k = 0.0367, k = 0.0649, Da = 1, Db = 0.5 [3]. We defined delta t to be 1 and used the default 2D Laplacian functions.

We used our color binning algorithm on an image of a real cheetah skin to generate a color palette of the image's top four dominant colors. Four was chosen because it seemed like a rough approximation of the number of distinct colors present in the real cheetah skin image. We then passed in the generated color scheme along with Karl Sim's parameter values as inputs into Gray-Scott model. Lastly, we assigned each cell a color based on its percentage of chemical A within the cell. Specifically, we manually defined four values in [0, 1] that each belong to one of the four dominants colors. Chemical A values that fall in between any of the 4 values are assigned an interpolated color.

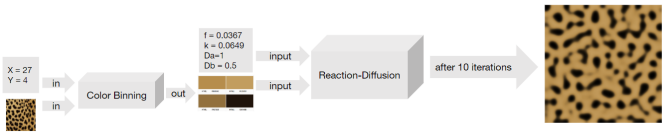

Fig. 11. Process for generating cheetah skin using Gray-Scott reaction diffusion.

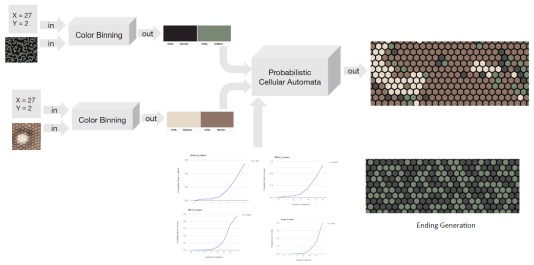

Fig. 12. Process for generating lizard skin.

10 PROBABLISTIC TOTALISTIC CELLULAR AUTOMATA

To implement the ocellated lizard skin pattern described in the next section, we used a probabilistic totalistic cellular automata model [4]. The implementation of the model is very similar to the 3-dimensional outer totalistic cellular automata model described in section 3. The main difference between these two models lies in the definition of their rule sets. In outer totalistic CA, if a cell has the number of neighbors required to continue living or be reborn, it will do so. In probabilistic totalistic CA, meeting the number of neighbors requirements to continue living or be reborn does not guarantee a change in state. Instead, each change in state is defined by a probability distribution.

11 OCELLATED LIZARD SKIN

To simulate an ocellated lizard skin, we implemented a hexagonal probabilistic CA model with probability distributions based on the distributions depicted in the paper, *A living mesoscopic cellular automaton made of skin scales* [4]. However, we had to make several modifications to the probability distributions in the paper because the paper was a biological study of real lizard scales (e.g. the scales weren't perfectly hexagonal as depicted in our CA model). For example, the probability distributions used in the paper had assigned probabilities for scales with up to 7 neighbors, while our CA model is limited to the 6 neighbors corresponding to the number of sides of a hexagon. To convert the paper's probability distribution to one we could use, we simply removed the probability corresponding to 7 neighbors.

Furthermore, the research paper only focused on the transition of green to black and black to green states. An ocellated lizard has brown/white skin at an early age, and then the skin transforms to black/green when the lizard is fully grown. Thus, the paper did not consider the early stages of color in a ocellated lizard skin, namely brown to white and white to brown. Since we want to visualize these color transformation as well, we came up with our own transition probabilities for those colors. We defined the probability distribution of white to brown to be the same probability distribution as black to green, and the probability distribution of brown to white had a similar distribution to black to green.

We randomly initialized our CA to start with 60 percent brown and 40 percent white scales, and allowed the brown and white scales to start turning into green or black after the 100th iteration of the algorithm. This also meant we had to define additional probability distributions for white to black/green and brown to black/green. The probabilities we used were not in the research paper, and we set based on the realism of the visually generated instances.

Finally, we applied the color binning algorithm to two input images (the brown-white image represents the color of lizard skin at an early age, while the green, black color is the lizard's skin color when the lizard is fully grown). Using these color schemes and the probability distributions, we were able to simulate the appearance of ocellated lizard skin over time.

12 CONCLUSION AND FUTURE WORK

To summarize, our project can be divided up into two main goals/parts. We first wanted to make it easier to visualize the different generations present in one instance of Cellular Automaton (CA). Our second goal was to develop a procedure to generate skins by leveraging the biologically plausible mechanisms of CA and RD and a color binning algorithm.

We hope to extend these findings to other classes of CA, because application of color to CA has not been explored in depth, especially with three-dimensional CA. The properties and applications described can likely be easily extended to two-dimensional totalistic CA and possibly other classes of CA.

For example, the generations a cell was first born in, as visualized using the palettes described previously, can be applied to two-dimensional totalistic CA in an analogous way (and so too can be extended to other dimensions).

Furthermore, we would like to explore more applications of color palettes to CA. It would be interesting to explore properties such as the density of a certain rule, for example. This would be done by assigning a certain color to a cell based on the number of neighbors it has, or some similar method. In the future, we would also like to explore how to describe the symmetry of CA in some way using colors.

There is also the problem, in the three-dimensional case especially, where it is difficult to see a majority of the cells because they are blocked by others closer to the viewer. We hope to somehow resolve this issue in some way using properties like the opacity of the cells, for example.

Regarding our work on procedurally generated animal skin, we would like to further explore other algorithms that are known to mimic skin patterns. We would also like to discover new parameter values in existing models (e.g. Gray-Scott) that could mimic other animals.

13 CONTRIBUTIONS

- Kenny Chen: Wrote the code for the color picker, the interpolation algorithms, and for some of the 3D CA. Helped write final report and prepare final presentation deck.
- Randy Fan: Wrote code for all parts except for the interpolation methods. Researched all project topics and created checkpoint reports and summary decks. Put together the 2-minute video. Helped write final report and prepared final presentation deck.

REFERENCES

[1] Author, *LaTeX: Title*. Other, Publication, Year.
[2] https://en.wikipedia.org/wiki/Pastel_(color)
https://www.karlsims.com/rd.html
[4] https://www.nature.com/articles/nature22031extended-data
[5] https://phys.org/news/2012-09-cheetah-stripesa-genetic-tale.html
[6] http://math.bu.edu/people/heijster/PRESENTATIONS/HEIJSTER_RPI2011.pdf
[7] https://www.semanticscholar.org/paper/Evolution-of-natural-patterns-from-random
[8] http://www.stephane-gobron.net/Core/Publications/Papers/2006_ACRI06.pdf
[9] https://plus.maths.org/content/spotting-lizards
[10] http://tetrahedral.blogspot.com/2011/08/cellular-automaton.html