Ȧdea - Evolving glyphs for aiding creativity in typeface design*

Daniel Lopes
University of Coimbra, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering
Coimbra , Portugal
dfl@dei.uc.pt

João Correia
University of Coimbra, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering
Coimbra, Portugal
jncor@dei.uc.pt

Penousal Machado
University of Coimbra, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering
Coimbra, Portugal
machado@dei.uc.pt

ABSTRACT

Being creative in Graphic Design often requires protected experimentation processes. We present an evolutionary engine for generating glyphs, aiding designers to explore during the creative process. The system employs a Genetic Algorithm to evolve SVG using both interactive and automatic fitness assignment. We present topological variation operators that promote the exploration of adequate topologies and we compare the performance of topological and conventional operators for generating uppercase "A"s. The results demonstrate that the topological crossover operator performs more efficiently both regarding fitness and phenotypes.

CCS CONCEPTS

• **Human-centered computing → Systems and tools for interaction design**;

KEYWORDS

generative design, generative typography, computational creativity

ACM Reference Format:
Daniel Lopes, João Correia, and Penousal Machado. 2020. Ȧdea - Evolving glyphs for aiding creativity in typeface design. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389964

1 INTRODUCTION

Novelty is one of the fundamental characteristics of describing creativity [1]. Yet, finding novel aesthetics often requires a protracted process of exploration (trial and error) of solutions or tools. Numerous Computational Creativity (CC) applications have been successful in the generation of art [4], music [5] and design [3], by applying Evolutionary Computation (EC) or Machine Learning (ML) techniques. Yet, several (CC) systems, mainly the ones based on ML, often end up creating imitations of existing styles . On the other hand, we argue that EC may have greater potential to find novelty due to their similarity to human design processes — search the unexplored space of possibilities, often with a specific conceptual target limiting the possibilities —, and allowing the exploration of a higher number of possibilities.

In this paper, we present a system for evolving variations of SVG glyphs. The goal is to generate ideas for designers to create novel glyphs or typefaces. We evolve SVG rather than raster images due to: (i) endless resizing (convenient for real GD applications); (ii) easier post-production; (iii) easier to create digital typefaces once these are natively represented in vectorial formats such as SVG. Furthermore, the work explores the creation and testing of topological variation operators which meant to preserve the topology of the glyphs.

2 APPROACH

Glyph evolution was implemented using a genetic algorithm in JavaScript. A genotype consists of an SVG path, composed of a variable number of "line" (L) and "move to" (M) points in a 100x100 pixels canvas. A phenotype consists of the SVG render of its genotype.

Any character may be evolved. For creating an initial population, a set of typefaces are randomly picked out of 977 *Google fonts*. For each selected typeface, a respective SVG glyph is generated using *opentype.js*[1]. For starting from randomly defined individuals, the points of each SVG are randomized within the SVG canvas.

The conventional crossover operator is based on standard 2-point operators [2]. Our crossover operator, which we refer to as topological crossover, intends to shift points that are topologically similar. i2's points are sorted by its Cartesian distance to each point of i1. Thereafter, a point from i1 is more likely to crossover with its closer points from i2. The selection chances decrease exponentially as i2's points are further away in the sorted set of points.

Five mutation methods were implemented and run as follows: (i) random point deletion; (ii.a) random point translation; (ii.b) translation of a random array of consecutive points (iii) type toggling of a random point; and (iv) random point creation (L or M). Evaluation may be performed interactively by clicking over phenotypes or automatically assessed by a pre-trained neural network —*Tesseract.js*[2]. Further parameterization may consulted in *https://cdv.dei.uc.pt/adea.*

Regarding selection, tournament and elitist selection [2] were implemented. The evolutionary process may be finished manually by pressing a button, or automatically when a set number of generations was run or a set percentage of the population is fitted.

*For trying the latest version of the system please visit *https://student.dei.uc.pt/~dfl/Adea.*

[1] *https://github.com/opentypejs/opentype.js*
[2] *https://tesseract.projectnaptha.com*

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07…$15.00
https://doi.org/10.1145/3377929.3389964

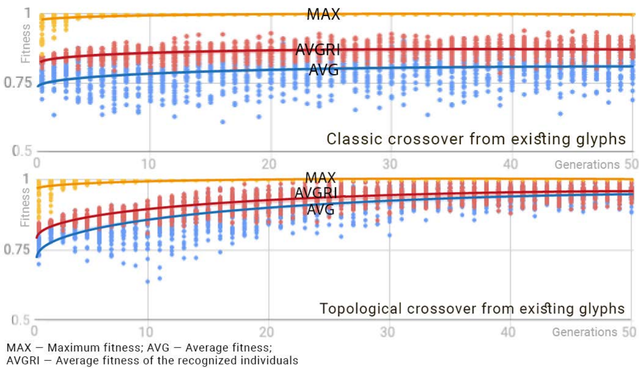

Figure 1: Conventional *vs* topological crossover from populations of existing glyphs (30 seeds of 50 generations).

3 EXPERIMENTAL SETUP AND RESULTS

Experiments were conducted by evolving uppercase "A"s using both conventional, both topological, and isolating each operator, either from existing and random populations. 30 seeds per setup were ran. The remaining parameters were fixed as follows: (i) population size of 50 (ii) elitist selection with a tournament size of 2; (iii) evolve until 50 generations; (iv) 80% chance to perform crossover; (v) 5% chance for each mutation operator to occur per gene; (vi) a maximum translation distance of 20 pixels; (vii) 95% chance for a new point to be L rather than M; (viii) a maximum of 20% of an individual may crossover/mutate per iteration; (ix) 80% chance to shift with closer points on topological crossover; (x) automatic fitness assignment; (xi) a target of 80% confidence (the glyphs may representative of the character yet these may differentiate from the training examples of *Tesseract*); (xii) 0.5% confidence margin (fitted above 0.995); (xiii) only individuals bigger than 5% of the canvas were considered representative.

Noteworthy results were observed while comparing the crossover operators starting from existing glyphs (see Fig. 1). Using the topological operator, both AVG and AVGRI maximized faster. The difference between AVGRI and AVG (AVGDIF) drastically minimizes, suggesting that more individuals were recognizable "A"s. Contrarily, the conventional operator tends to maintain a lower AVGDIF along generations. In accordance to such, the conventional operator generated considerably prickly and barely readable phenotypes for most seeds. The topological operator often generated readable glyphs, made of more regular shapes.

For demonstrating the feasibility of the system regarding other setups, we refer to Fig. 3 which showcases other characters evolved from existing glyphs using automatic evaluation and regarding different generations. We suggest visiting *https://cdv.dei.uc.pt/adea* for assessing how *Ȧdea* may be used to produce proper GD artefacts.

4 CONCLUSION

We presented *Ȧdea*, a system for evolving variations of SVG glyphs, which may serve as a starting point to explore new design spaces. Topological variation operators were tested by evolving uppercase "A"s and using *Tesseract.js* for fitness assignment. The experimental results indicate that the topological crossover operator outperforms the conventional one in the considered experimental settings, particularly when applied to individuals with a meaningful topology, such as existing font sets. Future work will focus on: (i) exploring different automatic evaluation methods; (ii) including figurative SVG images; (iii) (iv) testing the system through a user survey; (v) generating whole typefaces.

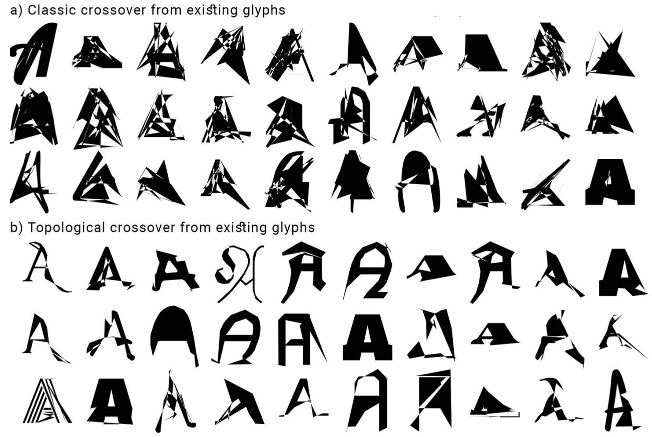

Figure 2: Conventional *vs* topological crossover from populations of existing glyphs (best fitted individual of each seed)

Figure 3: Other characters evolved from existing glyphs using topological methods, automatic evaluation and regarding different generations. For more examples, please visit *https://cdv.dei.uc.pt/adea.*

5 ACKNOWLEDGMENTS

This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020 and is partially supported by *Fundação para a Ciência e Tecnologia*, under the grant SFRH/BD/143553/2019.

REFERENCES

[1] Margaret A Boden. 1996. Creativity. In *Artificial intelligence*. Elsevier, 267–291.
[2] A E Eiben and James E Smith. 2015. *Introduction to Evolutionary Computing* (2nd ed.). Springer Publishing Company, Incorporated.
[3] S. Kitamura and H. Kanoh. 2011. Developing Support System for Making Posters with Interactive Evolutionary Computation. In *2011 Fourth International Symposium on Computational Intelligence and Design*, Vol. 1. 48–51. https://doi.org/10.1109/ISCID.2011.21
[4] Penousal Machado, Tiago Martins, Hugo Amaro, and Pedro H. Abreu. 2014. An Interface for Fitness Function Design. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design – Third International Conference, EvoMUSART 2014, Granada, Spain, April 23-25, 2014. Proceedings (Lecture Notes in Computer Science)*, Juan Romero, James McDermott, and João Correia (Eds.), Vol. 8601. Springer.
[5] Asterios Zacharakis, Maximos Kaliakatsos-Papakostas, Costas Tsougras, and Emilios Cambouropoulos. 2018. Musical blending and creativity: An empirical evaluation of the CHAMELEON melodic harmonisation assistant. *Musicae Scientiae* 22, 1 (2018), 119–144. https://doi.org/10.1177/1029864917712580