Which type is your type?

Jéssica Parente[1] and **Tiago Martins**[1] and **João Bicker**[1] and **Penousal Machado**[1]
[1]University of Coimbra, CISUC, DEI
{jparente, tiagofm, bicker, machado}@dei.uc.pt

Abstract

An interactive evolutionary system to generate letterings is presented. The system allows the creation of a wide range of alternative designs that can be used as stimuli for inspiration or for the creation of visual identities with different variations. This work began as a parametric system that generated glyph designs by recombining parts of skeletons extracted from existing typefaces which are then filled with custom shapes. In this paper, we employ a Genetic Algorithm to evolve the input parameters of this parametric system. The experimental results show that the presented evolutionary system enables users to interactively create unique letterings according to their aesthetic preferences.

Introduction

As human beings, the way we communicate is one of the most unique characteristics that define us. According to Cheng (2006), typography is the visual manifestation of language and the instrument that turn characters into words and words into messages. In the modern world, typography is also a way of “given meaning”, it needs to transmit resonance and depth to the messages it is transmitting. To communicate a message, a designer can use the composition of the elements and typography. Therefore, the design of type can be useful to add layers of meaning (Cheng 2006; Shaughnessy and Bierut 2009).

With the technological revolution, type design tools have changed. Now, typefaces have to be optimised taking into account where they’ll be read and the target audience. However, as Adrian Schaughnessy (Shaughnessy and Bierut 2009) points out, today’s type designers continue to do what they have always done: they are changing and adapting to developments in technology, media and literacy. The technology made possible new ways of exploration and allowed the type designer to explore previously unthinkable fields. Consequently, through these new possibilities, more typefaces emerged, but also the uncertainty of their quality. Moreover, with the emergence of Artificial Intelligence (AI), in the twentieth century, the potential of machines to be creative in their way can now be explored by academics and practitioners from diverse disciplines. In the typography field, appeared useful tools that provide a wide variety of alternative designs that promote new ideas during the design process. These new tools offer good support in the design process which can be an advantage when compared with conventional tools. By taking advantage of these computation systems we can find inspiration and unlock a creative block. However, these systems should respect some typographic rules by creating a balance between what the user can change and what the system automatically performs. With that in mind, we decided to create an Interactive Evolutionary Computing (IEC) system that generates letterforms, providing a wide range of alternative designs that can be used as stimuli for inspiration or even for the development of visual identities with different variations (Lupton 2006; Shaughnessy and Bierut 2009; Lehni 2011).

This project began with the creation of a less sophisticated system that designed glyphs by the combination of skeletons of existing typefaces and posterior filling. To develop the system three aspects were worked out: (i) the development of the structure of the typefaces generated and the codification of the different elements of the structure of the letter in different layers; (ii) the combination of layers of different typefaces; and (iii) the creation of glyphs through the generation/ modification of the elements of these layers. The IEC system uses the drawing process of the parametric system and evolves the glyph design parameters. An example of a generated lettering can be found on Figure 1. Each population is composed of letterings, so each individual represents a sequence of glyphs. We implement a Genetic Algorithm (GA) to evolve a set of parameters that control the generation of letterings. Then, we created a graphic user interface to allow the interactive guidance of the evolution process.

The remainder of this paper is organised as follows. Related Work Section presents related design projects in the domains of type design and IEC systems. Approach Section describes the Parametric and the Evolutionary System. Experimentation Section validates and demonstrates the potential of our system as a computer-aided creativity tool and discusses the achieved results. Finally, the Conclusion and Future Work section summarises our work and presents future research directions.

Related Work

In the early 1990s, the evolution of software to design fonts opened doors to new methods to create type. The Beowulf font (FontFont nd) appeared alongside with the first series

Proceedings of the 11th International Conference on Computational Creativity (ICCC’20)
ISBN: 978-989-54160-2-8

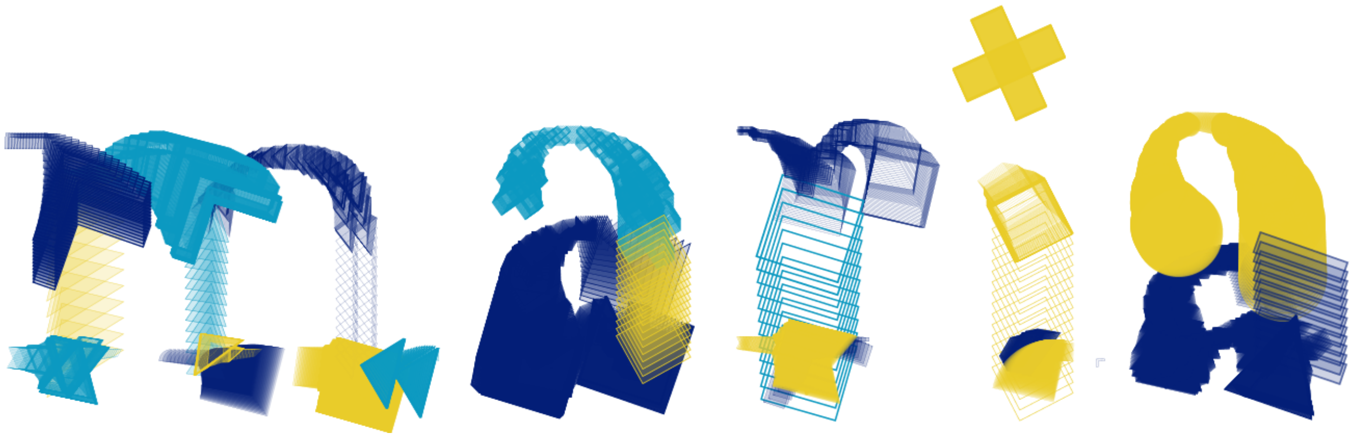

Figure 1: Example of a lettering evolved with the presented system.

of fonts with random outlines and programmed behaviour. They make use of a kind of pre-programmed randomness, they randomly move the points of the contours to deform the glyphs and, due to this there are not two identical glyphs. Today it is also possible to create glyphs that change according to data. Typography Music (Silanteva 2011) is a system that generates glyphs that react to music. The glyphs are formed from a grid and constructed by the combination of layers. Each layer is constituted by a range of modules and the shape of each module changes with the type of music, for instance for an organic sound the modules are circles, for an analogue sound, they are octagons.

With these new possibilities, computational creativity becomes a topic of discussion and computational systems that try to imitate human creativity appear. In 1992, Gary McGraw and Douglas Hofstadter proposed (Rehling and Hofstadter 2004), a system for the automatic generation of the lowercase letters of the roman alphabet in different but internally coherent styles. Starting with one or more seed grid-based letters, the system attempts to create the rest of the alphabet in such a way that all letters share that same style. Nowadays, we also see an increasing attempt to humanised computational intelligence. Interactive Evolutionary Computation (IEC) is one of these research domains and it is used in diverse research categories, for instance, design and computer-generated animation, music, face image generation, speech processing, image processing, among others. IEC as an optimisation method that involves Evolutionary Computation (EC) and it optimises a target system based on a user's subjective evaluations. EC systems had the advantage of providing a wide range of alternative designs that can be used as stimuli for inspiration. Besides, when we use an IEC system, we can blend the capabilities of EC optimisation with human evaluation and make fuller use of both of them. An experimental evaluation involving two types of interaction styles — Direct Manipulation and Interactive Evolutionary Design — to do creative tasks in a type design system can be found in (Lund 2000). The goal of this research was to compare these two kinds of interactions and it was found that that direct manipulation prototype offers a higher degree of freedom to design typefaces. Direct Manipulation is more suitable when the objective is clearly defined. On the other hand, Interactive Evolutionary Design proved to be the interaction where the users were more active; and it is more suited for creative tasks. Good solutions can be explored with the use of IEC systems; they can adapt, select and create even "better" solutions from a generation to another. With that in mind, Jaksa Kuzma and Sincak created a system (Kuzma, Jakša, and Sincak 2008) that helps the user to create typefaces. The user's evaluation affects the evolving process. The structure of the typefaces is based on Computer Modern font and it has several parameters to change the design of the glyphs. Alphabet Synthesis Machine (Levin, Feinberg, and Curtis 2001) is another system that has the same goal, the system creates abstract alphabets from a writing simulation using a Genetic algorithm (GA) — a search heuristic that is inspired by Charles Darwin's theory of natural evolution. The developed algorithm evolves a population of candidate glyphs according to a set of fitness metrics established by the user. The developed glyphs evolve as individuals to improve their characteristics, and as a species. Genotyp (Schmitz 2004) is another similar system, it generates typefaces by combining genetic characteristics of different fonts. The system allows the combination of different fonts and manipulation of their genomes. The combination of different fonts can result in mutations, but the granted inheritance can be modified later manually (Schmitz 2004). Evotype (Martins et al. 2015; 2016; 2018) is a project, divided into iterations, that explores different ways of designing glyphs. In the first iteration, the glyphs were designed by the combination of line segments arranged in a rectangular grid. Then, they were evaluated according to the visual similarities they had concerning the previously selected font (Martins et al. 2015). In the second iteration, glyphs were designed by the combination of shapes inserted by the user. To the evaluation part, they used the evaluated system of the last iteration combined by a classification model trained with numerous existing typefaces (Martins et al. 2016). In the last iteration, a stencil approach was created in which they generate stencils composed of line segments. This iteration allows the design of all letters in a more coherent way (Martins et al. 2016; 2018). Some approaches explore EC systems of type design creation outside the Roman alphabet. (Fischer 2004) is a tool that supports genetic operators by the addiction of variations to the design of the fonts based on Bézier curves. (Unemi and Soda 2003) is another system for font design that uses IEC technique, the system creates Japanese Katakana from very simple stroke elements. To create the genome of each individual they encoded some parameters for drawing elements. The initial population has sixteen individuals with random genes and the user can breed the font that he/she liked most.

Approach

In this paper, we present an IEC system that generates letterings. Our goal for the system was to create something that could be useful for designers to develop glyphs and letterings. The first part of this section presents the initial stage of this project, a parametric system to generate typefaces. The second part describes an evolutionary system which evolves the parameters of the previous glyph design system. We also created a graphic user interface to help the user to control the IEC system. We present the system, its possibilities and, in the end, we validate it.

Parametric System

This project began with the creation of a parametric system to generate typefaces. The system creates glyphs by the extraction of the skeleton of existing typefaces and separation of the skeletons into parts. The skeleton of each generated glyph will be the result of the combination of parts of the skeletons of different typefaces. Then, the generated skeleton is filled by the drawing of shapes repeatedly all over the skeleton. For more information, you can consult the first iteration of the parametric system on (Parente, Martins, and Bicker 2018).

Developing the structure of the glyphs For us, the structure of the generated glyphs was a import part to take into account. Nowadays, several typographic and generative systems develop typefaces, but most of them mostly focus on the letters' filling and use, for the structure, hand-drawing typefaces which are mostly static. Our first major goal for this project was the creation of the structure of the glyphs. For that, we decided to use existing typefaces and extract their skeletons. Therefore, we make sure that the design of the glyphs follows the rules of traditional font design.

There are diverse works that explore skeletons extraction through the use of different methods these days (Naccache and Shinghal 1984; Gonczarowski 1998; Dimauro, Impedovo, and Pirlo 2011). We decided to use the Zhang-Suen Thinning Algorithm (Zhang and Suen 1984) that aims to extract the structural lines of a binary image. This algorithm receives an image consisting only of pixels of two different colours, for example, black and white, and returns it modified. All the pixels that are not essential to understand the image are removed, which is ideal for this work, where we need to extract the skeleton of typefaces.

Once the possible solution for the skeleton extraction was found, we decided to implement it in our project. The skeleton extraction process begins with the scanning of all pixels. If the evaluated pixel fits a series of conditions is defined as white. The process repeats over time and ends when none of the pixels suffers any alteration.

Division of the skeleton into strokes With the skeletons extracted, we needed to find a method that identified the different parts. After analysing the generated skeletons, we noticed that when a point was part of three segments it divides different strokes of the skeleton. A stroke is composed by at least two points, but in general, they are more, and a set of strokes compose a skeleton. To test this theory, we scanned all the points of the skeleton and when we found a border point, we created a new stroke, and so on till the end. Figure 2 presents the process of skeleton extraction and division into parts of a glyph for the letter "h". The circles highlight the border points and a different colour was applied to differentiate the different parts detected.

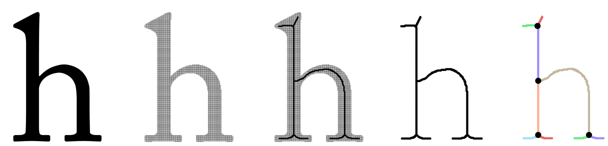

Figure 2: Process of extracting a glyph skeleton.

Skeletons recombination One of our goals was to combine different typefaces' structure into a glyph. Since we already had a system that generated skeletons, the next step was the combination of parts between different skeletons. To each glyph we needed to assess which stroke of each skeleton could be associated with each other. To do that, we determined the angular velocity and the central point. Therefore, each stroke did not need to be equal to the other to be pairing. To put it in context, angular velocity is a vector that represents the process of changing the orientation of a given line. For a line, the value will be equal to 0 and it will increase as it becomes more curved. The central point is an average of all points of the part in question.

The pairing process started with the ordering of the strokes of the first skeleton, from the longest to the shortest length. In principle, the error that could arise from the combination would be minimised. Then, each of these parts was compared with all the constituent parts of the second skeleton and so on. When the compared parts had a similar angular velocity and centre point, they were considered corresponding and we moved on to the next skeleton. At the end of the cycle, we had, for a given character, several versions for each stroke.

Give body to the skeletons In the section *Developing the structure of the glyphs* we mentioned that the pairing process started with the exclusion of the pixels furthest from the centre of the stroke. Therefore, when calculating the distance between each pixel of the skeleton to the nearest pixel from the border, we determine the width of the original typeface. With this measure, we can replicate the glyph, or increase or decrease the weight proportionally. Then, using different shapes (e.g. circles, triangles, squares, or other abstract shapes) repeatedly we fill the strokes of the final skeletons and generate typefaces using different colours and transparencies. The filling of each stroke is composed of modules repeated along the stroke line. To each generated typeface, we could determine which typefaces we wanted as input and we can choose the colours and modules to use.

Evolutionary System

Although the parametric system was already capable of generating typefaces, we wanted to explore the system more. Each generated glyph had a series of parameters that still had a lot to explore. Besides, parameters such as the density of shapes repeated in the filling were not even used as a variable. Thus, a system to evolve the input parameters of the parametric system can provide us with new ideas for the creation of the glyphs by the generation of unpredictable designs that could be something useful as stimuli for inspiration. With that in mind, we decided to employ an IEC system that generates letterings. To achieve this, a GA is implemented to evolve different populations of letterings which are based on the designing process of the parametric system developed earlier.

Representation The evolutionary system evolves a population of letterings. Therefore, each individual represents a sequence of glyphs. The genotype of each individual consists of a list of tuples containing integers. Each tuple represents a stroke of a glyph. Each integer encodes the index of a setting for a given attribute for the stroke that is represented by the enclosing tuple. The attributes of each stroke (see Figure 3) are the input typeface, the used shape, the shape's scale, the number of shapes or density, the shape's colour, the opacity of shape's fill, the opacity of shape's contour, the skeleton's opacity and the width of the shape's contour. The filling of each stroke is composed of the repetition of shapes along the stroke line. The phenotype of each individual is the lettering generated with the parametric system already described using the settings encoded in the genotype.

Crossover, mutation and evaluation The generated glyphs are the result of the user's subjective evaluations. During the evolutionary process, the users choose their preferred letterings from the current population. This leads to the creation of more populations containing individuals with visual properties that go according to their taste. To do so, we use the individuals selected by the users and then we apply genetic variation operators, including crossover and mutation. The crossover operator breeds the selected individuals by recombining their genetic information. Then, the mutation operator permits the variation of genes in the genotype.

When the user selects multiple individuals, we apply the crossover operator to random pairs of these individuals and then apply the mutation operator to the resulting offspring. When the user selects only one individual, we use the mutation operator to create variations of it. When the user selects no individuals, we apply this procedure to the last selected

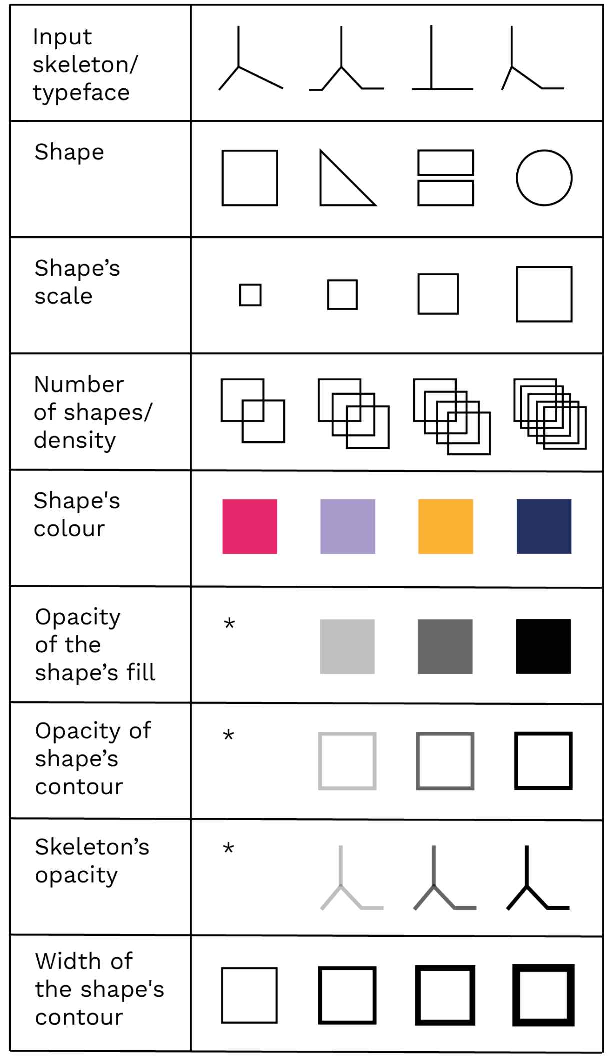

Figure 3: The attributes of each glyph stroke. The visual elements marked with * are not visible due to their low opacity.

individuals, which may not necessarily be of the latest generation.

Visualisation We developed a graphic user interface to enable the user to visualise, select and breed letterings (Figure 4). The individuals (letterings) of the current generation are arranged on a grid to facilitate users' choice. The evolutionary process begins with the selection of the values to each parameter by the users. More specifically, they can choose the possible values for the skeleton, shape, scale, density, colour, fill-opacity, contour opacity, skeleton opacity and contour width. Additionally, users can also write the letters that will compose the evolving lettering. During the evolutionary process, users can interactively pick the lettering(s) that they liked the most. Also, at any moment of the evolutionary process, users can export the evolved letterings as vector files to, for example, apply them in a design process or make further design refinements.

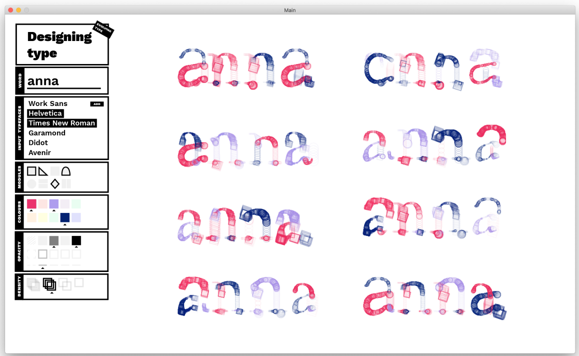

Figure 4: Screenshot of the evolutionary system.

Experimentation

As previously mentioned, our IEC system allows the generation of very diverse letterings. This leads to numerous different possibilities to represent the same word. This occurs thanks to the big range of parameters that the system uses to create each glyph. In this section, we summarise the visual possibilities of this approach, wherein different parameter settings and analysed the results. During these experiments, we studied the impact of the parameter settings.

The values of each parameter are the following: (i) to the input skeletons we use eight typefaces as input (Didot, Times New Roman, Adobe Caslon Pro, Bodoni, Helvetica, Adobe Garamond Pro, Futura, Baskerville); (ii) to the shape parameter we decide to design 16 shapes; (iii) to the shape's scale, we established 4 values (from half to twice the size of the input typeface's width); (iv) to the density parameter, we established 6 proportional values to the font-size of the glyph (That way when the font-size is smaller the generated glyph had a lower level of detail); (v) we established ten possible colours to the shapes; (vi) to the opacity of fill, contour and skeleton we set 6 values (from 0 % to 100 % of opacity); and (vii) for the contour width we decided to use just 5 values. In the beginning, we established four possible values to the opacity of fill, contour and skeleton however, the probability of having a blank shape was too high. The users can determine which available values of each parameter to use. After some tests, we decided to keep the population size of 15 individuals and the mutation rate of 0.1.

According to the users' desire, diverse letterings can emerge by the alteration of the values for each parameter and posterior evolution of populations through the selection of favourite individuals. To understand the possibilities of the parametric system, we decided to explore the available parameters. Then, we ask two different users to evolved letterings are to validate the IEC system.

Generating Letterings

We began the exploration of our parametric system by the attempt to create glyphs similar to the common typefaces. We decided to use the word "maria" and limit the parameters to generate letterings with circles as the shape, black as the only colour and with 100% of opacity. Bodoni was the typeface chosen as the input to generate the skeleton and, to the scale, we use the scale of the original typeface. Figure 5 — Variation 1 shows the result. In the second variation of Figure 5, we let the system generate letterings with the same characteristics as the previous, but with more typefaces as inputs, typefaces that were similar to the one used above. In Variation 3 of Figure 5 we add more typefaces but more different than those used previously (Variation 4 of Figure 5) we use two very different skeletons, we use Bodoni and Helvetica typeface. By the observation of the letterings generated Figure 5, we can notice that the system can combine more than two typefaces and generate a third one by mixing their skeleton. We also see that the parts of each glyph are unique, even with the same parameters the system can generate variants.

Figure 5: Original glyphs (top) and variations (bottom) created by recombining strokes of the original glyphs with strokes of glyphs of other typefaces.

We also explored the variations on the shape's scale (Figure 6). We wanted to see the different behaviours of the system and their adaptation to different values of the shape's scale. We used just one typeface as input and we use the same black circles used previously and we employ opacities of 0 and 100% in the fill of the circles, Figure 6 and Figure 7 respectively. We noticed that the letterings generated in Figure 6 had more contrast, and for that reason, the differences between strokes became a lot more visible. Becomes interesting to observe the differences that the same lettering suffers only by changing the scale of the shapes. Besides, these variations in black and white with the opacity of the shape of 100% (Figure 6) are the ones that can be used more easily in a text, mostly the glyphs composed by strokes that have a width equal or less the original typeface used as input.

From our point of view, the contrast on the glyphs' skeleton present mostly on Variation 3 and 4 of Figure 6 can be very useful in the design context, use as data-driven logotypes or dynamic identities. On the other hand, Figure 7 presents a series of variations that had more detail, and that could be, more easily used to create an identity. Besides, the shades created by the overlap of the layers create another variant in the lettering.

Figure 6: Four variations in which we vary the shape's scale while maintaining the opacity of its fill set to 100%.

Figure 7: Four variations in which we vary the shape's scale while maintaining the opacity of its fill set to 0%.

In this system, we can also work with colours, and use different kinds of shapes in the design of a glyph. In Figure 8 and 9, we let the system generates letterings varying three parameters. In lasts variations, we decide to have a bigger level of abstraction in both Figure 8 and Figure 9. Each variation uses different values of scale, similar to the previous examples. The use of colours adds a new level to explore, besides the use of different opacities serves to mix different parts of the glyphs. The use of shapes without fill (Figure 9) adds much more levels of details, and it allows us to see all the shapes that compose the glyph. Thanks to the large variety of generated letterings they could be part of a composed of a dynamic identity with many variations.

Figure 8: Four variations in which we vary the shape, its colour and scale while maintaining the opacity of its fill set to 100%.

Evolving Letterings

In the last section we presented some of the visual possibilities of the parametric system. Now we pretend to demonstrate the potential of our IEC system applied to a design point of view. An analysis of the evolution of fitness across generations would be mostly pointless since we are not interested in demonstrating that genetic algorithms work (that has been established countless times before). Instead, we are interested in validating and demonstrating the potential of our system as a computer-aided creativity tool. For that purpose, we focus on the analysis of the results obtained by different users when working with the tool (Figure 10 and Figure 11).

The user A (Figure 10) decided to use the word "create". He uses all available shapes, typefaces and values of scale and density, but only four colours. The user ended up choosing letterings where the glyphs could be read perfectly. However, he believed that most of the generated individuals, even those never chosen, could be a possibility for application in an identity. Throughout the generations, several valid hypotheses were assumed by the user, but he wanted to test more to see where the system could take him. That is one of the advantages of this system. It is capable of generating non-expected versions that can unblock an artistic block.

Figure 9: Four variations in which we vary the shape, its colour and scale while maintaining the opacity of its fill set to 0%.

The user B (Figure 11) decided to use the word “Anna”. She used just part of the available shapes and colours and she decided to evolve letterings with skeletons composed with shapes without fill. Through the generations, the results were diverse, but the system ended up converging for the style she wanted, something more light and clean.

The diversity of the results highlights the expressive power of the tool and the impact of the user’s preferences on the outcomes of the system. It is totally possible that if the same users use the tool one more time with the same established parameters, they would create different generations. Another interesting thing about the evolutionary system is that it can create non-expected combinations. Nowadays, as designers we need tools to help us in the creative process, not to replace us. We need a combination between what the system generates to please us, according to our choices, and what the system does to create something more.

Conclusion and Future Work

This paper presents an IEC system that generates letterings. The system is composed by a parametric system that extracts the skeleton of existing typefaces, separate the skeletons into parts, recombine the parts in a final skeleton and fill the final glyph by the drawing of shapes repeatedly all over the skeleton. The second part of the system is an evolutionary algorithm that evolves parameters of the design of the glyphs. We demonstrate the system and its possibilities and validate them. Our main contributions include: (i) a parametric system capable of automatically create letterings; (ii) a evolutionary system that evolves the drawing parameters of the glyphs; (iii) exploration of different parameters of the design of a glyph into visual components; and (iv) an investigation into how evolutionary computation can be used in the field of type design.

Figure 10: Generation 1 (top) and 12 (bottom) of the evolutionary process guided by the user A. The selected letterings of the generation 12 are the ones that the user A liked the most.

It is important to highlight that it is a work in progress and for that reason, there is a set of things that we would like to do in the next iterations. In the actual system, the glyph that composed the letterings (individuals) are separated from each other. That means, that the glyphs of each word have no connection between them. In future work could be interesting to have some points in common in the glyphs of the same individual, at least in the glyphs corresponding to the same character. We also want to explore more the evolutionary by adding a system to save letterings that the user likes more. It would also be interesting that the system saves all the choices of the user. At this time, the system only looks for the lasts choices of the user. By making this alteration the system should adapt better to each user. In the future, we intend to conduct user studies in order to evaluate the evolutionary capabilities of the system, namely its ability to produce different and novel types that satisfy users’ preferences, as well as the system’s ability to promote users’ creativity leading them to explore new ideas and concepts.

References

Cheng, K. 2006. *Designing Type*. Laurence King.

Dimauro, G.; Impedovo, S.; and Pirlo, G. 2011. A new thinning algorithm based on controlled deletion of edge regions.

Figure 11: Generation 1 (top) and 8 (bottom) of the evolutionary process guided by the user B. The selected letterings of the generation 8 are the ones that the user B liked the most.

International Journal of Pattern Recognition and Artificial Intelligence 07.

Fischer, T. 2004. Cultural perception and technique as constraints in computational design evolution: A case study of generative chinese typography. *WSEAS Transactions on Information Science and Applications* 1:292–297.

FontFont. n.d. Ff beowolf. Retrieved from https://www.fontshop.com/families/ff-beowolf.

Gonczarowski, J. 1998. Producing the skeleton of a character. In Hersch, R. D.; André, J.; and Brown, H., eds., *Electronic Publishing, Artistic Imaging, and Digital Typography*, 66–76. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kuzma, M.; Jakša, R.; and Sincak, P. 2008. Computational intelligence in font design.

Lehni, J. 2011. Typeface as programme. Retrieved from https://www.typotheque.com/articles/typeface_as_programme.

Levin, G.; Feinberg, J.; and Curtis, C. 2001. Alphabet synthesis machine. Retrieved from http://www.flong.com/projects/alphabet/.

Lund, A. 2000. Evolving the shape of things to come - a comparison of interactive evolution and direct manipulation for creative tasks. In *Proceedings of the 3rd Generative Art Conference*. Milan, Italy: AleaDesign Publisher.

Lupton, E. 2006. *Pensar com Tipos*. Cosac Naify.

Martins, T.; Correia, J.; Costa, E.; and Machado, P. 2015. Evotype: Evolutionary type design. In Johnson, C.; Carballal, A.; and Correia, J., eds., *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, volume 9027 of *Lecture Notes in Computer Science*. Copenhagen, Denmark: Springer International Publishing.

Martins, T.; Correia, J.; Costa, E.; and Machado, P. 2016. Evotype: From shapes to glyphs. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16. New York, NY, USA: ACM.

Martins, T.; Correia, J.; Costa, E.; and Machado, P. 2018. Evotype: Towards the evolution of type stencils. In Liapis, A.; Romero Cardalda, J. J.; and Ekárt, A., eds., *Computational Intelligence in Music, Sound, Art and Design*, 299–314. Parma, Italy: Springer International Publishing.

Naccache, N. J., and Shinghal, R. 1984. An investigation into the skeletonization approach of hilditch. *Pattern Recognition* 17(3):279 – 284.

Parente, J.; Martins, T.; and Bicker, J. 2018. Generative type design: An approach focused on skeletons extraction and their anatomical deconstruction. In *Book of Proceedings of Typography Meeting*.

Rehling, J., and Hofstadter, D. 2004. Letter Spirit: A Model of Visual Creativity. In *ICCM*, 249–254.

Schmitz, M. 2004. genotyp, an experiment about genetic typography. Presented at Generative Art Conference 2004.

Shaughnessy, A., and Bierut, M. 2009. *Graphic Design: A User's Manual*. Laurence King Publishing. Laurence King.

Silanteva, D. 2011. Typographic music. Retrieved from http://www.ddina.com/index.php?/2011/typographic-music/2/.

Unemi, T., and Soda, M. 2003. An iec-based support system for font design. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics: Washington, D.C., USA, 5–8 October 2003*, 968–973.

Zhang, T. Y., and Suen, C. Y. 1984. A fast parallel algorithm for thinning digital patterns. *Commun. ACM* 27(3):236–239.

Acknowledgments

This work is partially 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 under the grant [SFRH/BD/148706/2019]